JASA/IPA共催セミナー(ET-WEST)

2011年6月16日

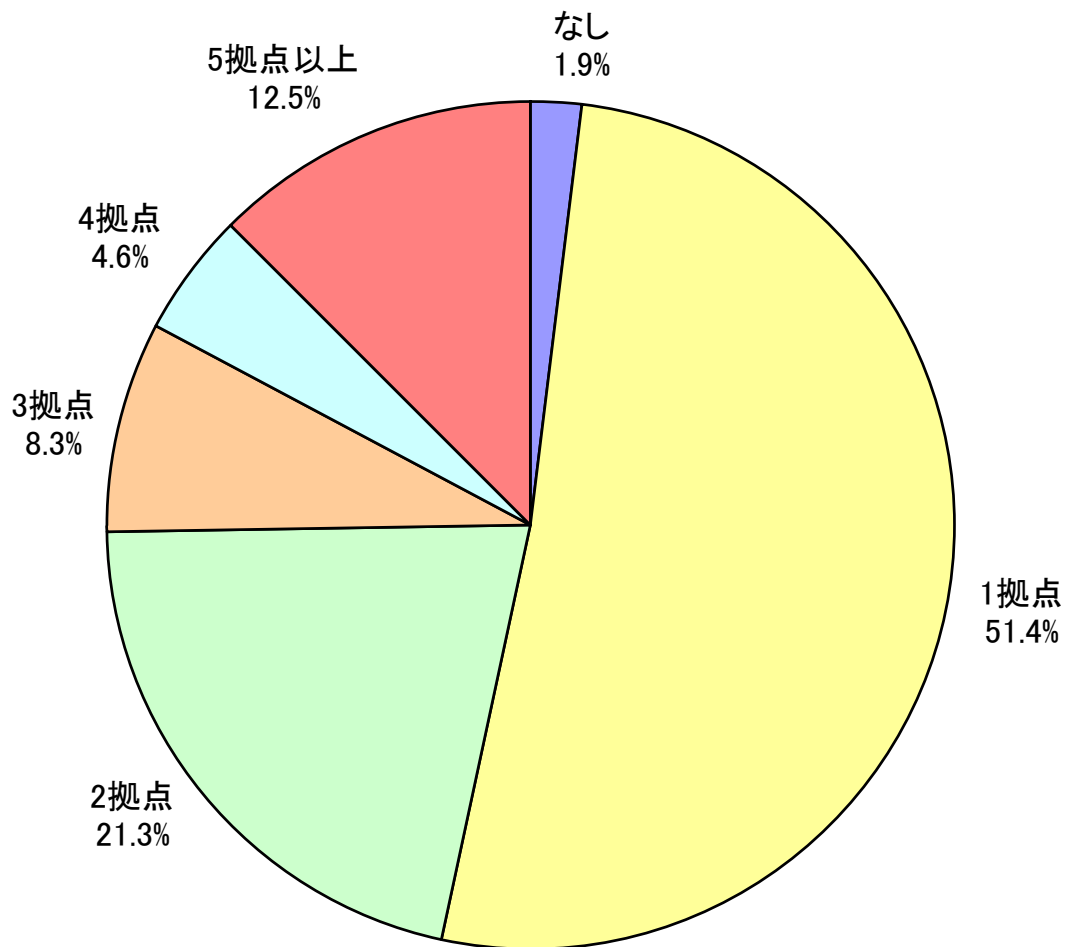
組込みソフトウェア向け開発プロセスガイド ESPR 解説

IPA 独立行政法人 情報処理推進機構
SEC ソフトウェア・エンジニアリング・センター

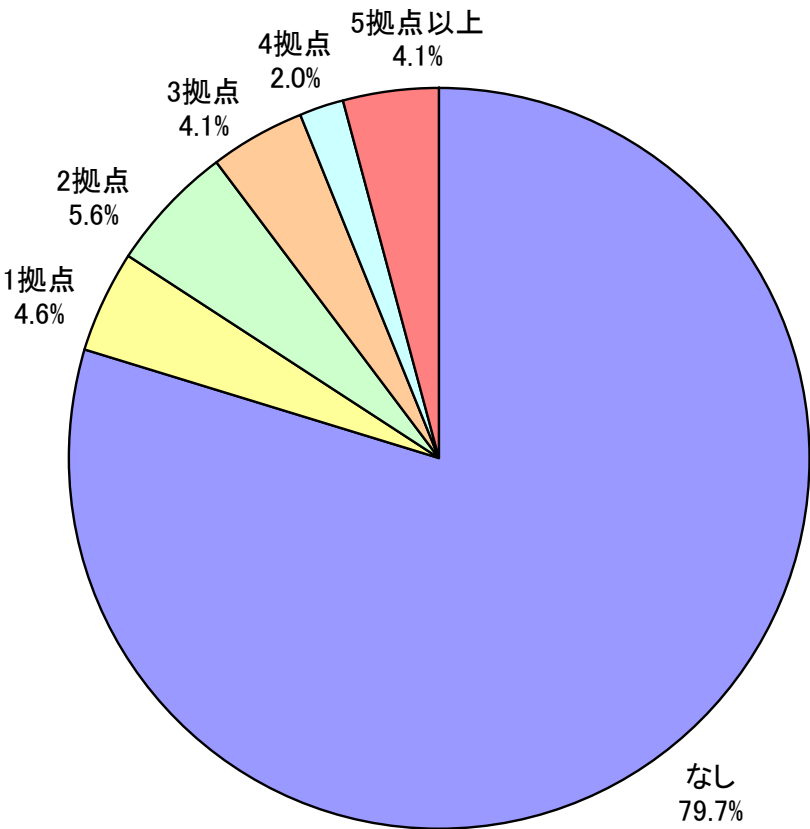
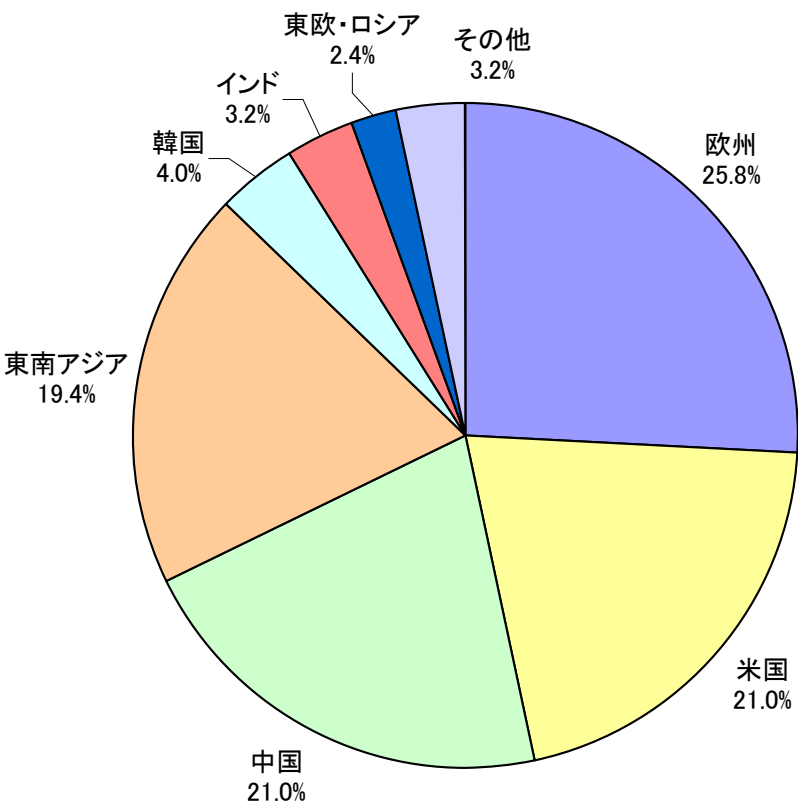
専門委員　村松　昭男

目次

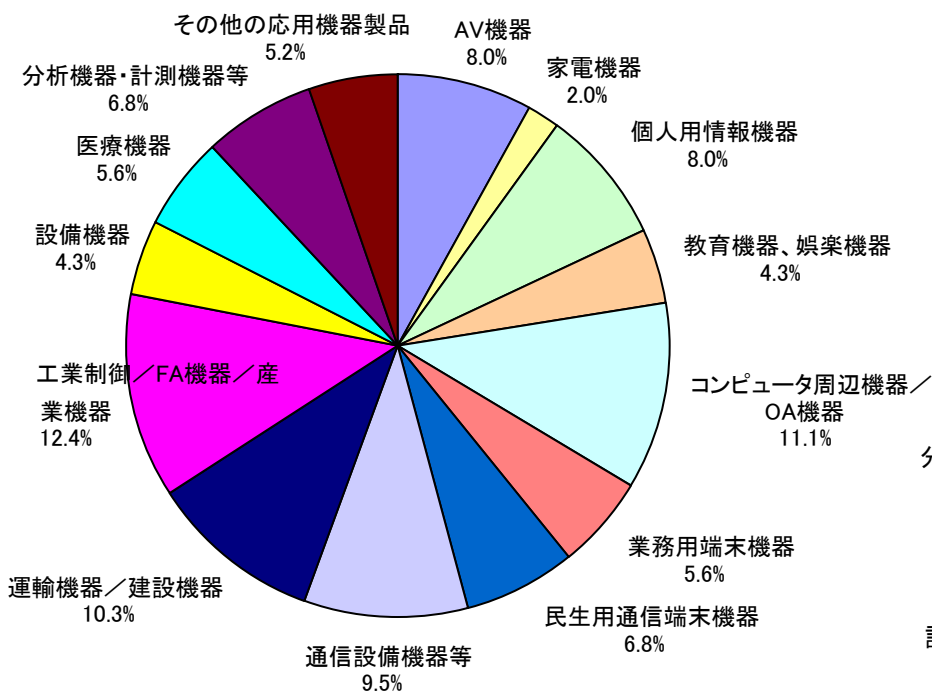
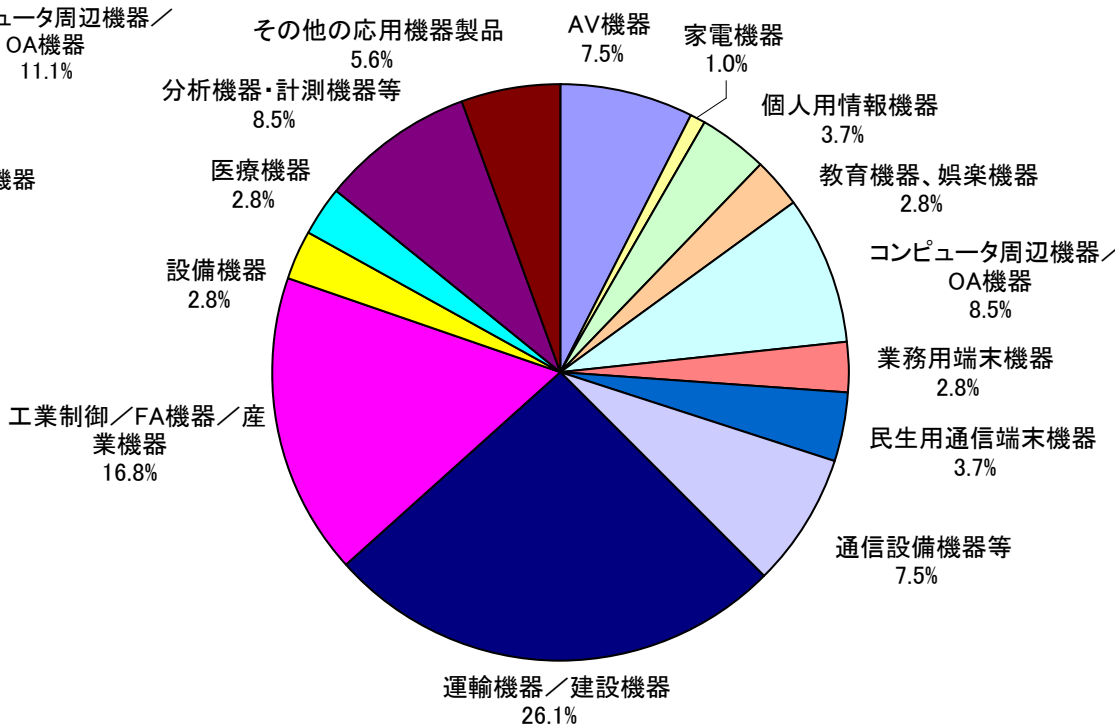
開発拠点（国内）

2011年版組込みソフトウェア産業実態把握調査報告書

国内開発拠点数

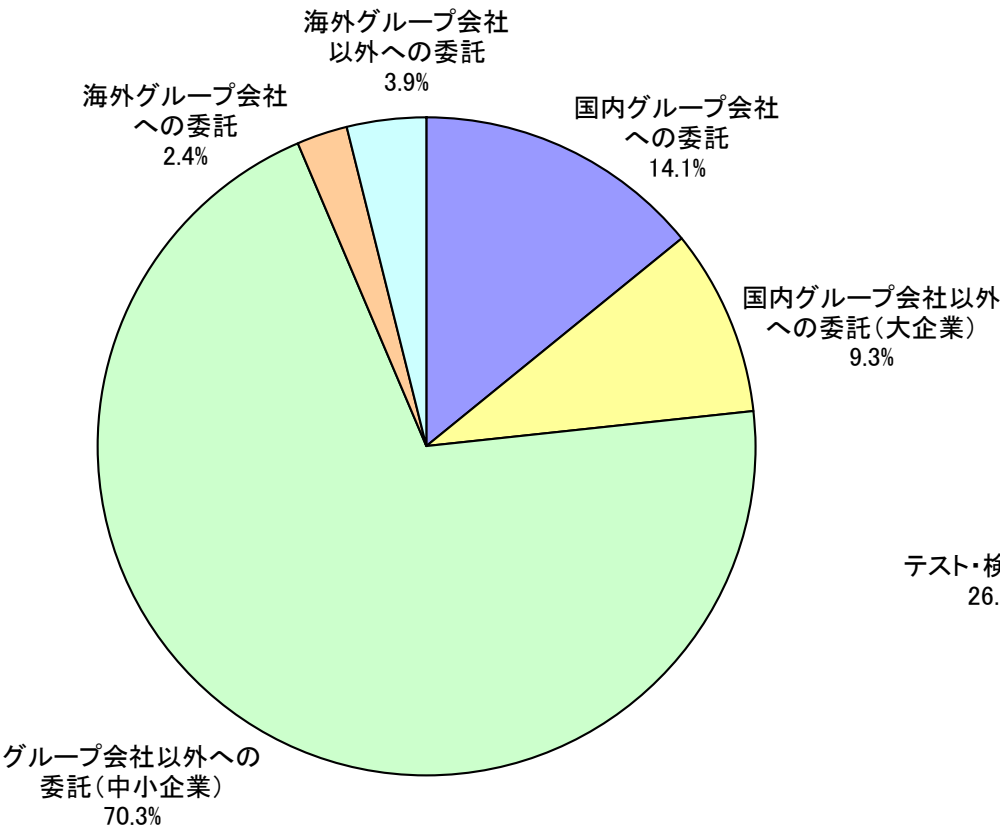
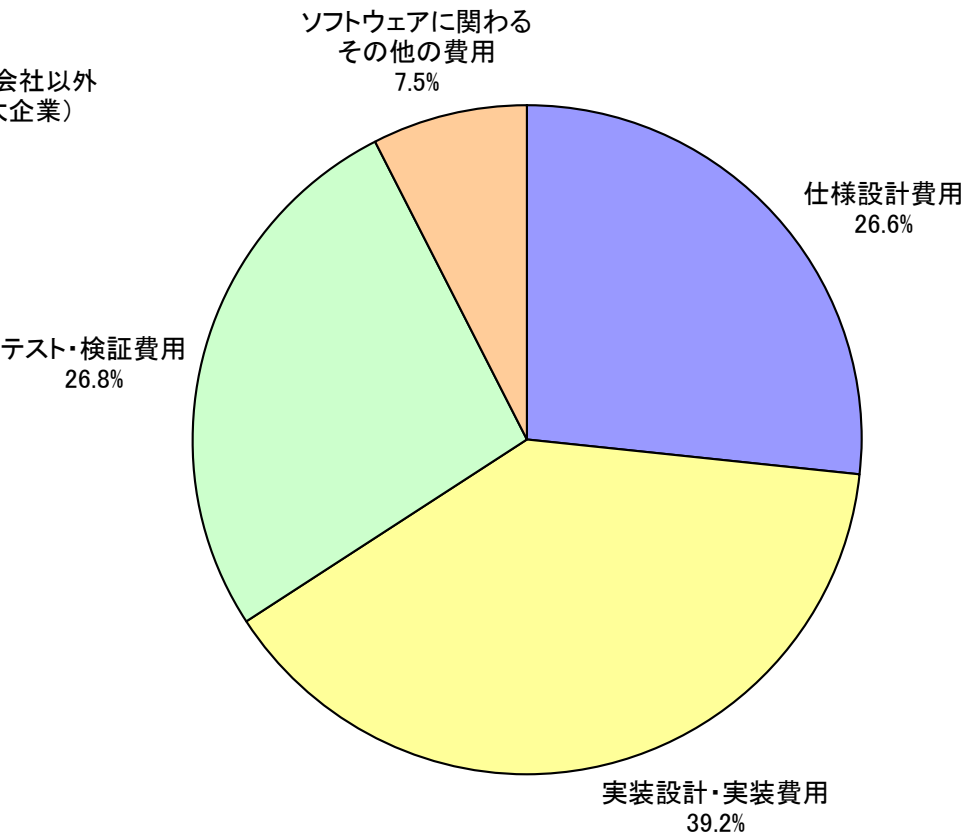
開発拠点（海外）

2011年版組込みソフトウェア産業実態把握調査報告書

海外開発拠点数

海外開発拠点所在地

事業カテゴリー

2011年版組込みソフトウェア産業実態把握調査報告書

主要事業のｶﾃｺﾞﾘｰ

規模の大きい開発事業

ソフトウェア費用

2011年版組込みソフトウェア産業実態把握調査報告書

委託先別費用

用途別費用

開発における課題

2011年版組込みソフトウェア産業実態把握調査報告書

開発課題

凡例：1番目　2番目　3番目

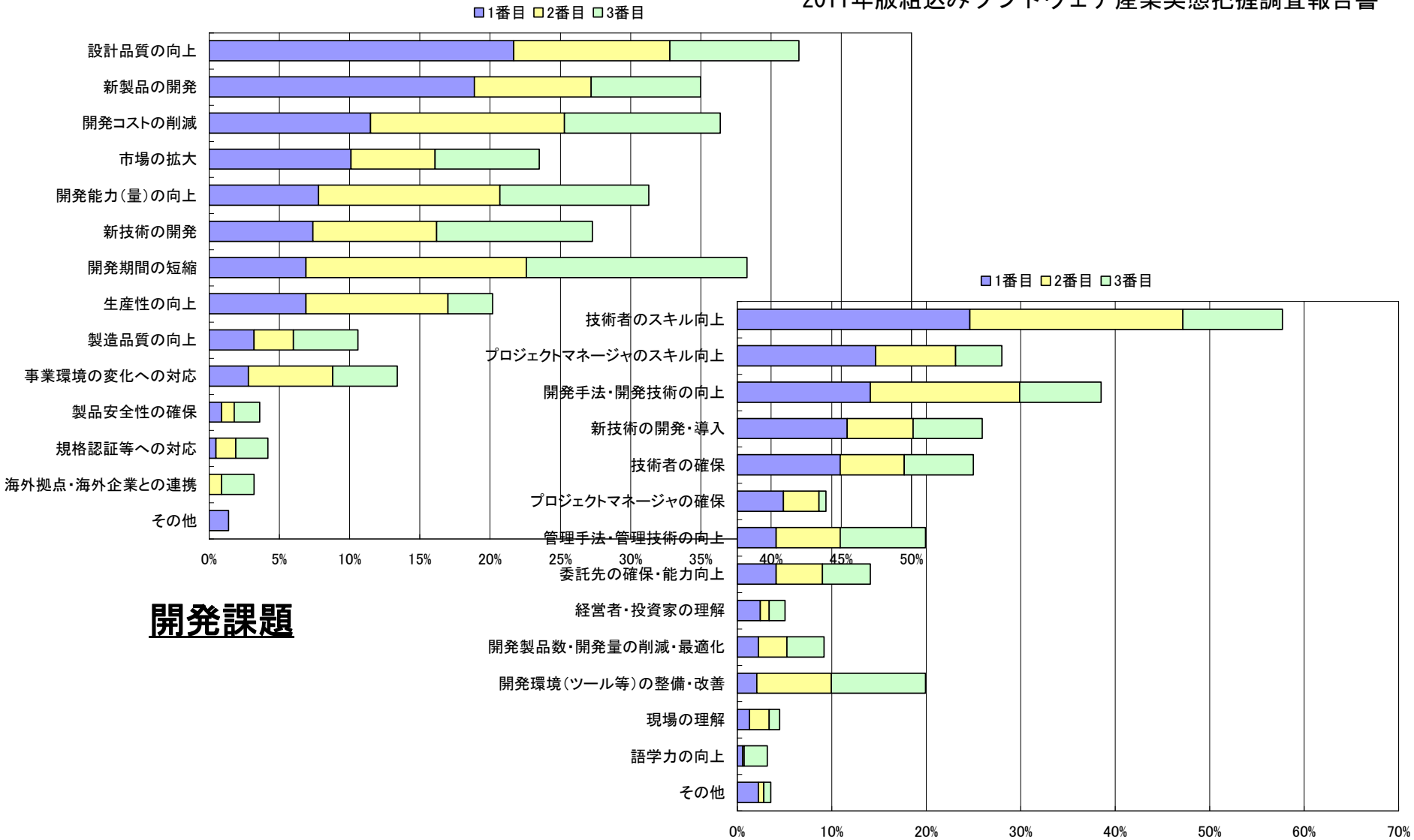
- 設計品質の向上
- 新製品の開発
- 開発コストの削減
- 市場の拡大
- 開発能力（量）の向上
- 新技術の開発
- 開発期間の短縮
- 生産性の向上
- 製造品質の向上
- 事業環境の変化への対応
- 製品安全性の確保
- 規格認証等への対応
- 海外拠点・海外企業との連携
- その他

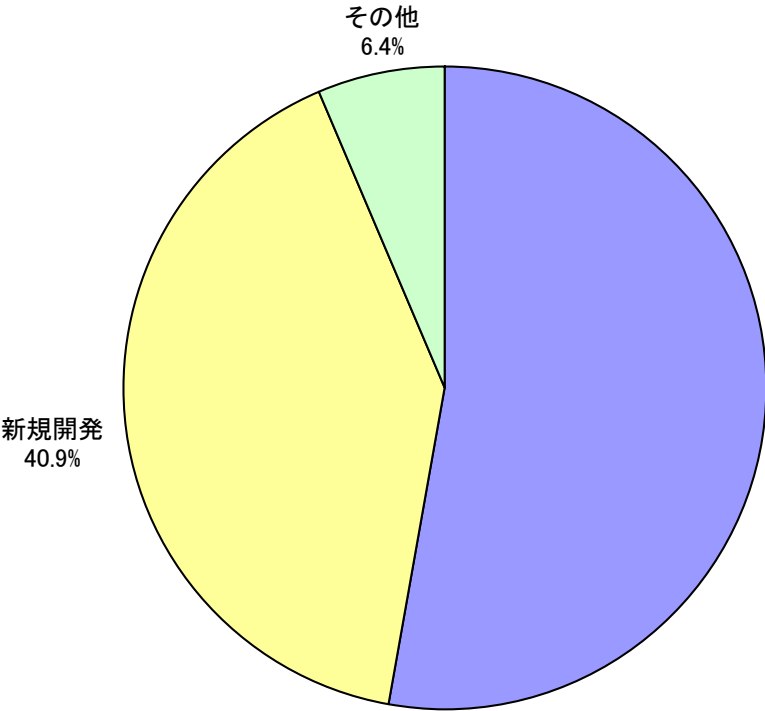
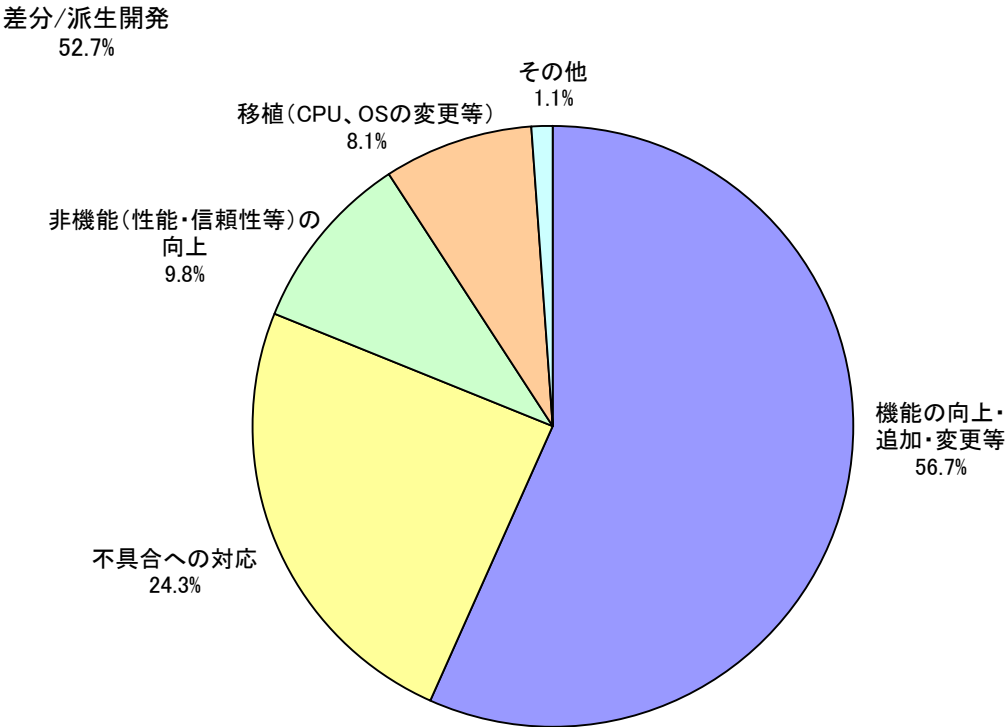
横軸：0%　5%　10%　15%　20%　25%　30%　35%　40%　45%　50%

解決手段

凡例：1番目　2番目　3番目

- 技術者のスキル向上
- プロジェクトマネージャのスキル向上
- 開発手法・開発技術の向上
- 新技術の開発・導入
- 技術者の確保
- プロジェクトマネージャの確保
- 管理手法・管理技術の向上
- 委託先の確保・能力向上
- 経営者・投資家の理解
- 開発製品数・開発量の削減・最適化
- 開発環境（ツール等）の整備・改善
- 現場の理解
- 語学力の向上
- その他

横軸：0%　10%　20%　30%　40%　50%　60%　70%

開発形態（プロジェクト件数）

2011年版組込みソフトウェア産業実態把握調査報告書

開発形態

差分・派生開発の内容

コード行数とプロジェクト件数

2011年版組込みソフトウェア産業実態把握調査報告書

1,000万行以上
1.2%

100万～1,000万行未満
4.4%

10万～100万行未満
18.4%

10万行未満
76.0%

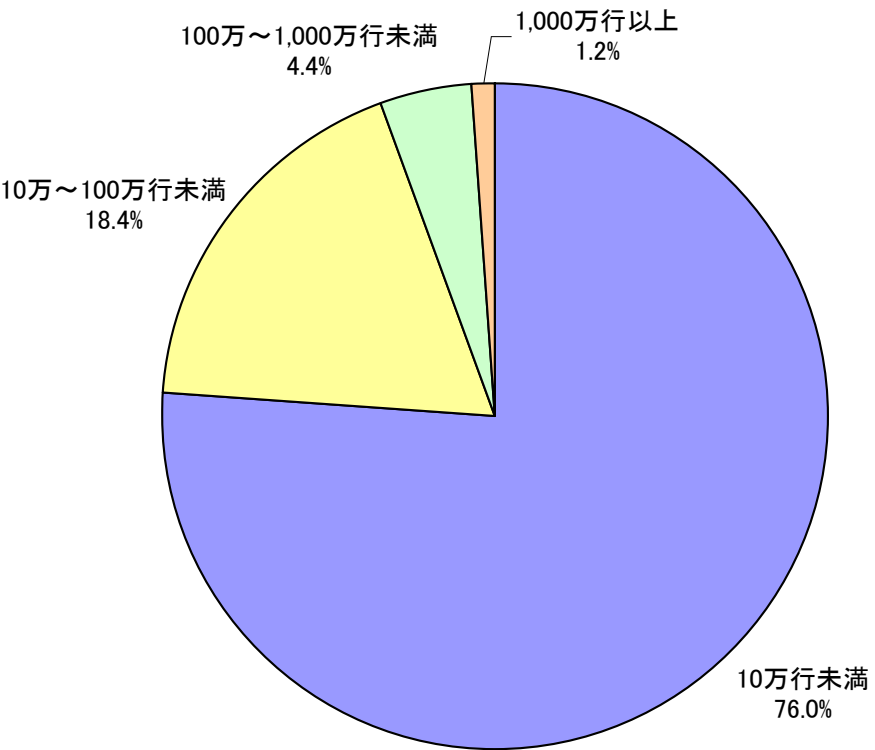
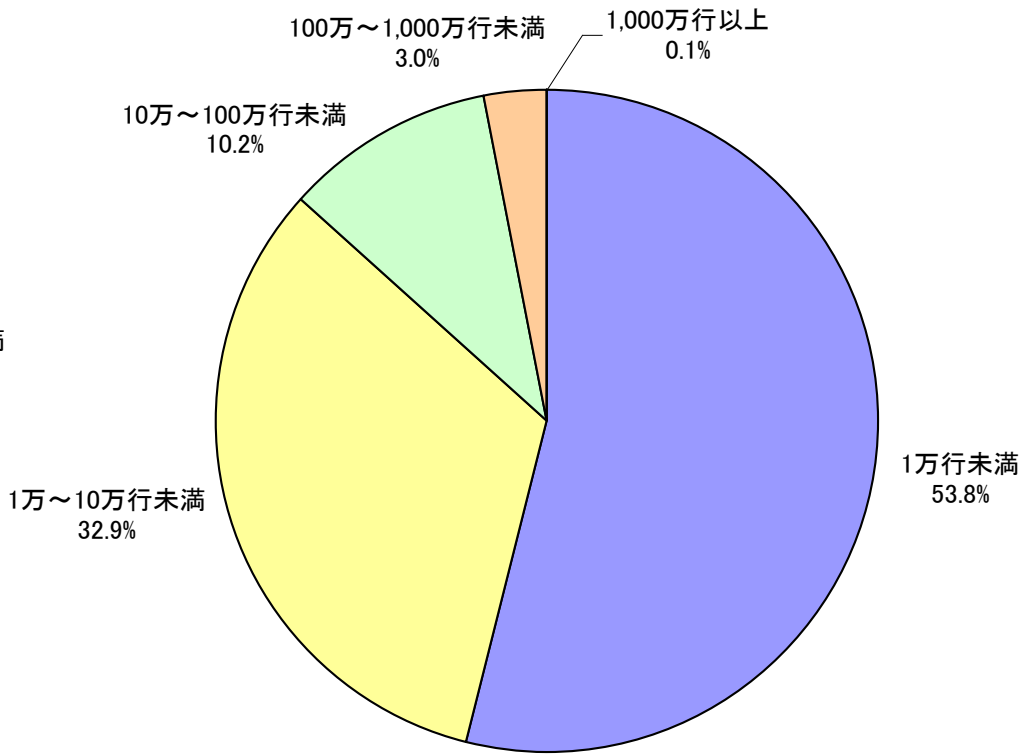
全行数別ﾌﾟﾛｼﾞｪｸﾄ件数

1,000万行以上
0.1%

100万～1,000万行未満
3.0%

10万～100万行未満
10.2%

1万～10万行未満
32.9%

1万行未満
53.8%

新規開発行数別ﾌﾟﾛｼﾞｪｸﾄ件数

コード作成方法

2011年版組込みソフトウェア産業実態把握調査報告書

ﾌﾟﾛｸﾞﾗﾑ作成方法

その他
0.4%

自動コード生成
8.1%

人手
91.5%

人手ｺｰﾄﾞ作成手段

アセンブリ言語
5.1%

その他
6.0%

Java
7.9%

C++/C#
23.7%

C
57.3%

自動ｺｰﾄﾞ生成手段

SysML
0.0%

その他
15.5%

形式手法系
（B、VDM等）
0.6%

XML系
1.6%

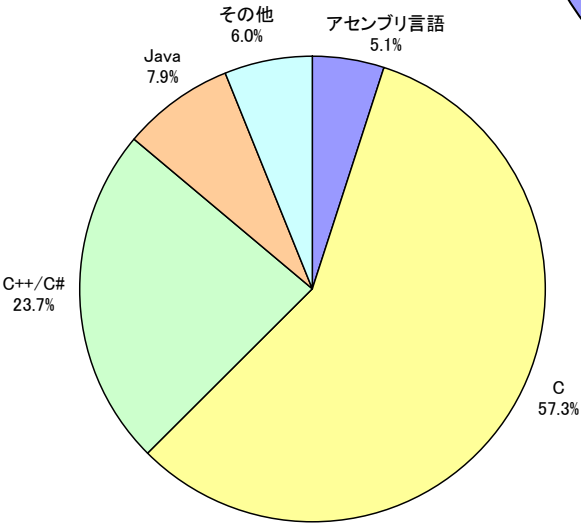
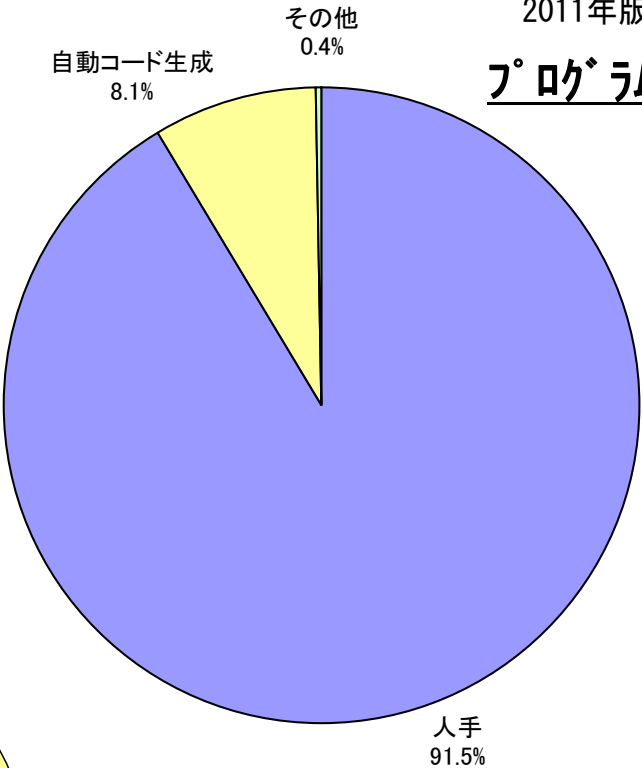
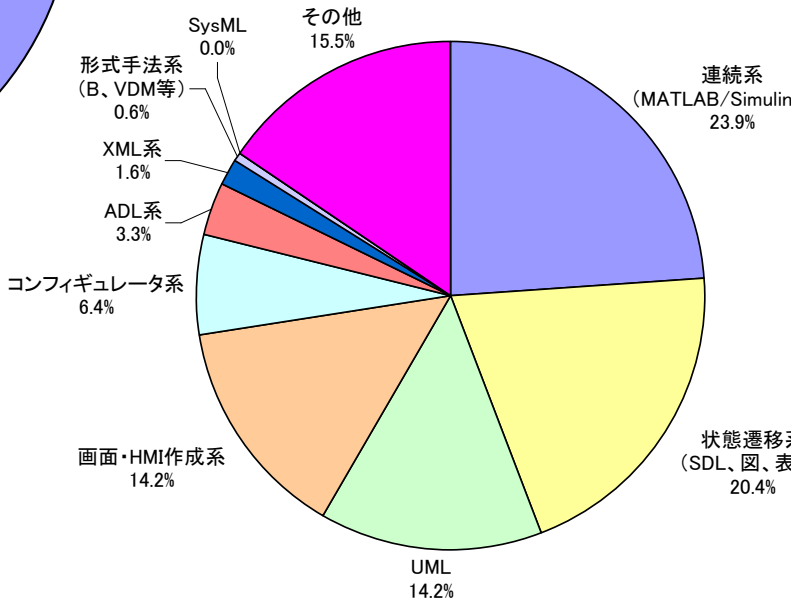
ADL系
3.3%

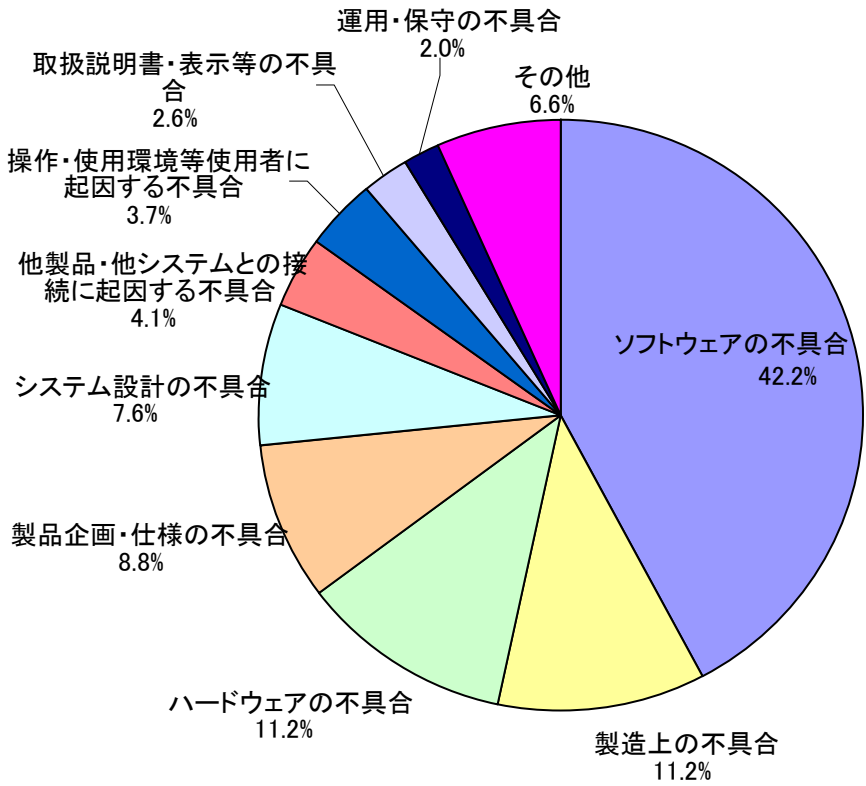
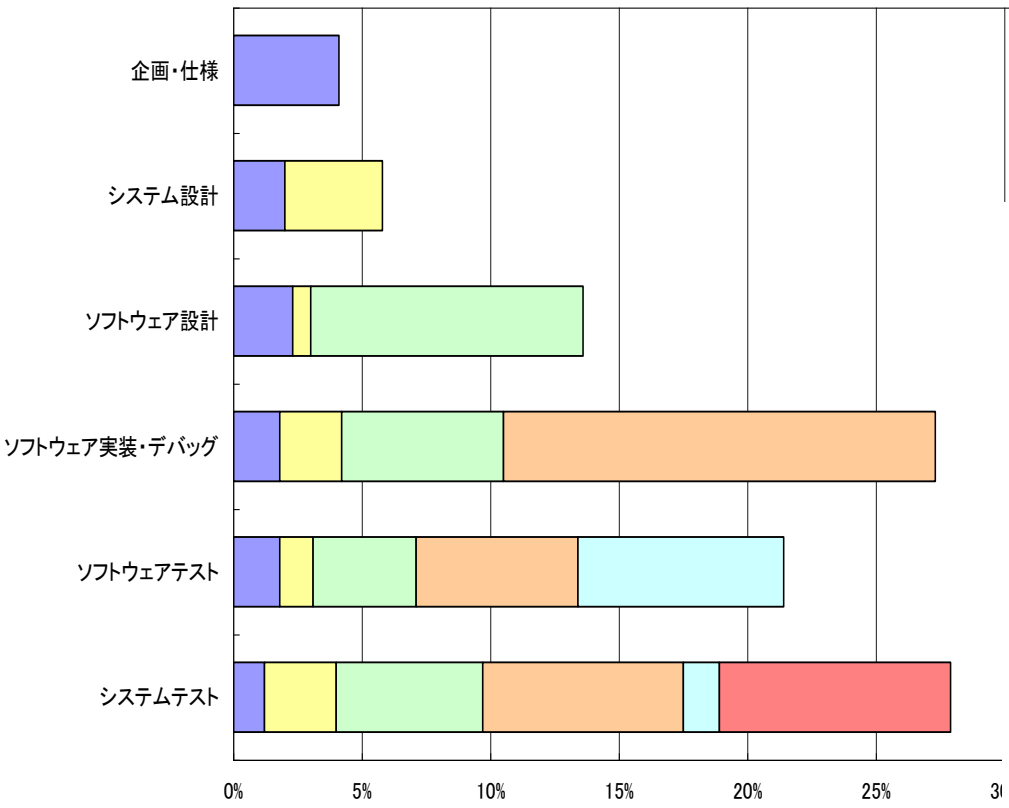
コンフィギュレータ系
6.4%

画面・HMI作成系
14.2%

UML
14.2%

連続系
（MATLAB/Simulin
23.9%

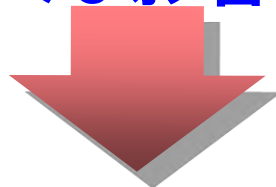
状態遷移
（SDL、図、表
20.4%

不具合発生状況

2011年版組込みソフトウェア産業実態把握調査報告書

不具合発見工程別不具合件数と発生工程

出荷後の不具合原因

品質に対する危機感

組込みシステムの不具合

✓個々の機器トラブルだけでは済まない
✓ネットワーク社会の中で，
より広範囲に多大な影響を及ぼすケースが増加

品質面でかなり気を使って開発する必要性

生命、財産、権利を守る

標準プロセスの制定・維持・定着が困難

外部起因の影響を受け易い

・ビジネス性：機能変更、仕様変更、出荷時期前倒し、装置出荷優先･･･

・ハード在りき：ハード原価削減、ハード開発/ソフト渡し遅れ･･･

企業･業種により開発手法が異なる

・自動車､家電､携帯電話､ｹﾞｰﾑ機器、航空機、ロケット･･･

・品質/安全･安心度合い、開発/保証期間、自社開発/共同開発等の相違

開発手法・技法が未成熟/発展中

・試行開発、職人肌の少数･精鋭、ﾐﾄﾞﾙ/ｱﾌﾟﾘｹｰｼｮﾝ･ｿﾌﾄｳｪｱの増加

・効率性/保守性よりも性能/コンパクトさ、設計よりも実装･･･

標準プロセスがない

・既存の標準(ISO12207,JIS160)はソフトウェアシステム向き

・具体的な内容となってない

ESPR基本コンセプト

- **やるべきことを明確化**
  （入力物、作業項目、出力物）
- **組込みソフトウェア開発を考慮**
  (他に、組込みシステムとしての開発作業、管理等の支援作業、安全性確保のための作業)
- **品質向上/確保のための仕組みにフォーカス**
  (入力物のチェック、出力物のレビュー)
- **柔軟な利用形態を提供**
  (入力/作業/出力をセットとした階層構造で部分利用も可能)
- **現場の具体的な情報/ノウハウを追加**
  （ドキュメントテンプレートの整備、具体的な実施内容、注意事項の記載）

やるべきことをやるのが基本

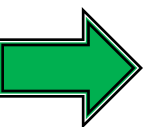
【背景】

大規模化 / 複雑化 / 短期化 / 高機能化 / 多様化 / 低コスト化

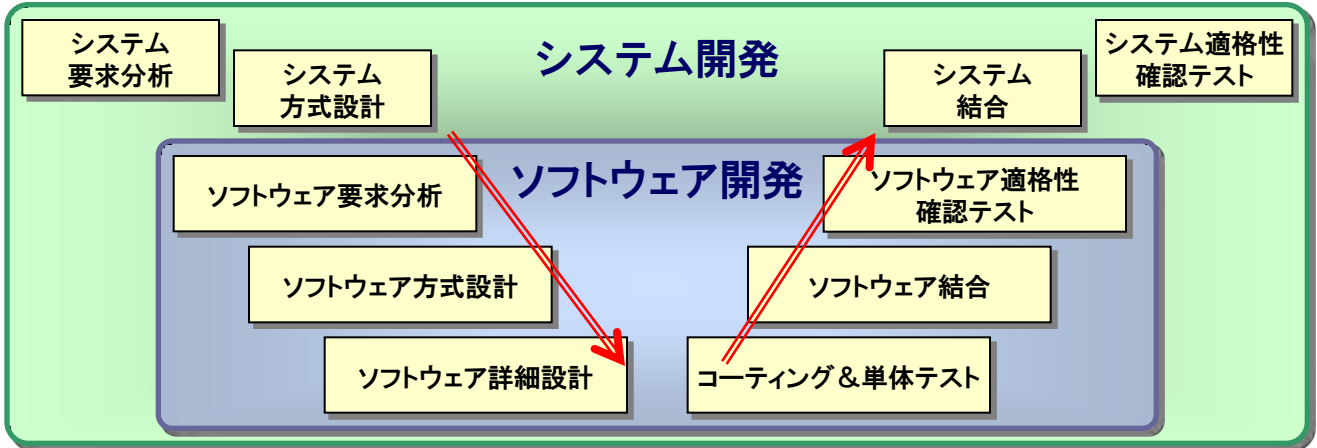
【現象】

品質問題 / コスト増 / 提供遅れ

【課題】・・・いろいろあるが

開発技術 / 要員スキル / 開発管理技術 / 品質管理技術 / ・・・

やるべきことを手順通り、手戻りなくチャンとやる

開発手順の遵守とIPOの明確化

■ 開発手順
(上流プロセス重視)

ISO/IEC 12207(JIS X 0160) 開発プロセスのアクティビティ

■ プロセスのIPO明確化
目的、入力・出力、
タスクを明確化

Input

システム要求仕様

ハードウェア仕様

(ドキュメント重視)

ソフトウェア要求分析

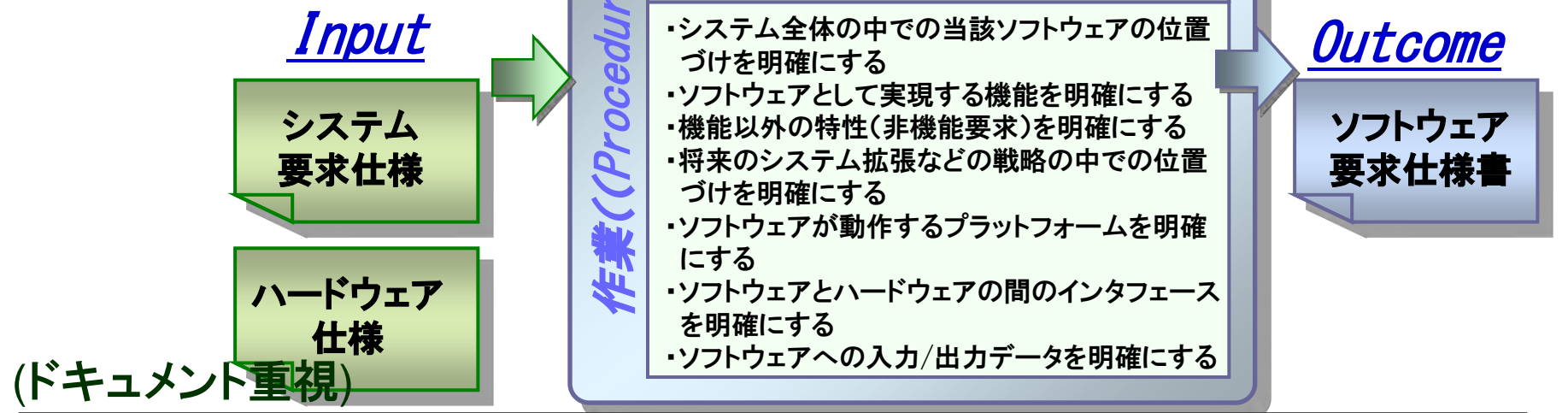
作業(Procedure)

Purpose
どのようなソフトウェアを開発するか明確にする

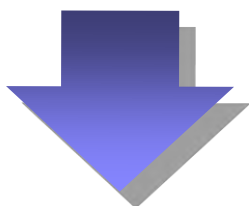
- ・システム全体の中での当該ソフトウェアの位置づけを明確にする
- ・ソフトウェアとして実現する機能を明確にする
- ・機能以外の特性(非機能要求)を明確にする
- ・将来のシステム拡張などの戦略の中での位置づけを明確にする
- ・ソフトウェアが動作するプラットフォームを明確にする
- ・ソフトウェアとハードウェアの間のインタフェースを明確にする
- ・ソフトウェアへの入力/出力データを明確にする

誰もが同じような
品質・生産物

Outcome

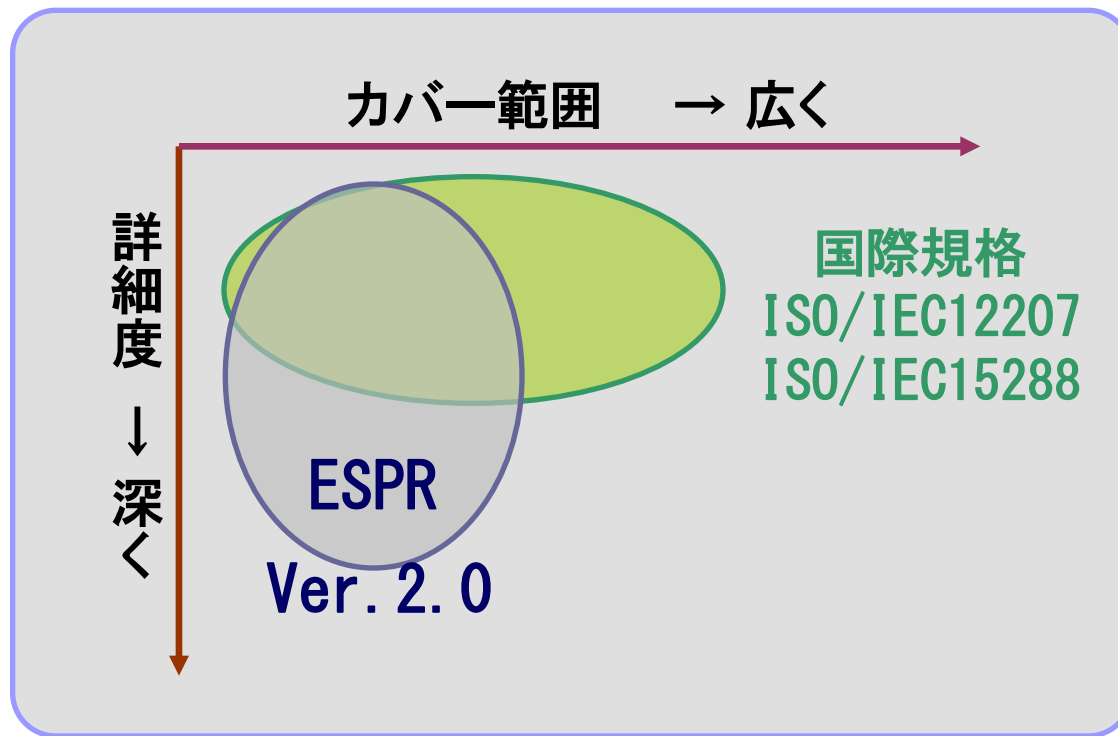
ソフトウェア要求仕様書

組込みソフトウェアの開発プロセスのお手本

ソフトウェア開発プロセスに関する標準類

ISO/IEC 12207 Software Lifecycle Process
ISO/IEC 15288 System Lifecycle Process
共通フレーム

組込みシステムに関する要素をブレンド

ESPR：
Embedded
System
development
Process
Reference

特徴
① 作業プロセス名の見直し
② 作業の入出力の整理
③ 作業上の注意事項の整理
④ ドキュメントテンプレート
⑤ 活用方法

国際規格とESPRの関係

組込みソフトウェア開発に最低限必要な作業の集まりとする
⇒横方向：プロセスのカバー範囲は国際規格ISO/IEC12207、15288よりは少ないプロセス
⇒縦方向：詳細度は組込みソフトウェア開発の実作業が見えるレベルの詳細な程度

ESPRのプロセス・カテゴリ

ESPR

システム・エンジニアリング・プロセス（SYP）

SYP: SYstem engineering Process

組込みソフトウェアが組み込まれて動作する組込みシステムとしてとらえた場合のシステム要求定義やシステム動作検証などの作業を整理したプロセス

ソフトウェア・エンジニアリング・プロセス（SWP）

SWP: SoftWare engineering Process

ソフトウェアとしての要求定義からソフトウェア総合テストまでソフトウェアを作る際の直接作業を整理したプロセス

セーフティ・エンジニアリング・プロセス（SAP）

SAP: SAfty engineering Process

安全・安心な組込みシステムを作り上げるために実施すべき作業を整理したプロセス

サポート・プロセス（SUP）

SUP: SUpport Process

ソフトウェア開発を円滑に進めるために必要となる支援作業や間接作業を中心に整理したプロセス

開発プロセスガイド全体図

プロセス定義

製品企画
V字モデル
製品審査
①詳細化の流れ
②確認の流れ

SYP：システム・エンジニアリング・プロセス
SYP1 システム要求定義
システム要求分析
システム適格性確認テスト
SYP4 システムテスト
SAP1 安全性要求定義
SAP2 安全性テスト
SYP2 システム・アーキテクチャ設計
システム方式設計
システム結合
SYP3 システム結合テスト
SAP：セーフティ・エンジニアリング・プロセス

SWP：ソフトウェア・エンジニアリング・プロセス
SWP1 ソフトウェア要求定義
ソフトウェア要求分析
ソフトウェア適格性確認テスト
SWP6 ソフトウェア総合テスト
SWP2 ソフトウェア・アーキテクチャ設計
ソフトウェア方式設計
ソフトウェア結合
SWP5 ソフトウェア結合テスト
SWP3 ソフトウェア詳細設計
ソフトウェア詳細設計
単体テスト
SWP4 単体テスト
SWP4 実装
コーディング
ハードウェア開発

ESPR x0180

SUP：サポート・プロセス
SUP1 プロジェクトマネジメント
SUP3 リスクマネジメント
SUP5 構成管理
SUP7 変更管理
SUP9 開発委託管理
SUP2 品質保証
SUP4 文書化と文書管理
SUP6 問題解決管理
SUP8 共同レビュー
SUP10 開発環境整備

ドキュメント・テンプレート

仕様書･設計書
テスト仕様書/報告書
内部確認レポート
共同レビュー記録
不具合管理票/台帳
プロジェクト完了報告書

活用編

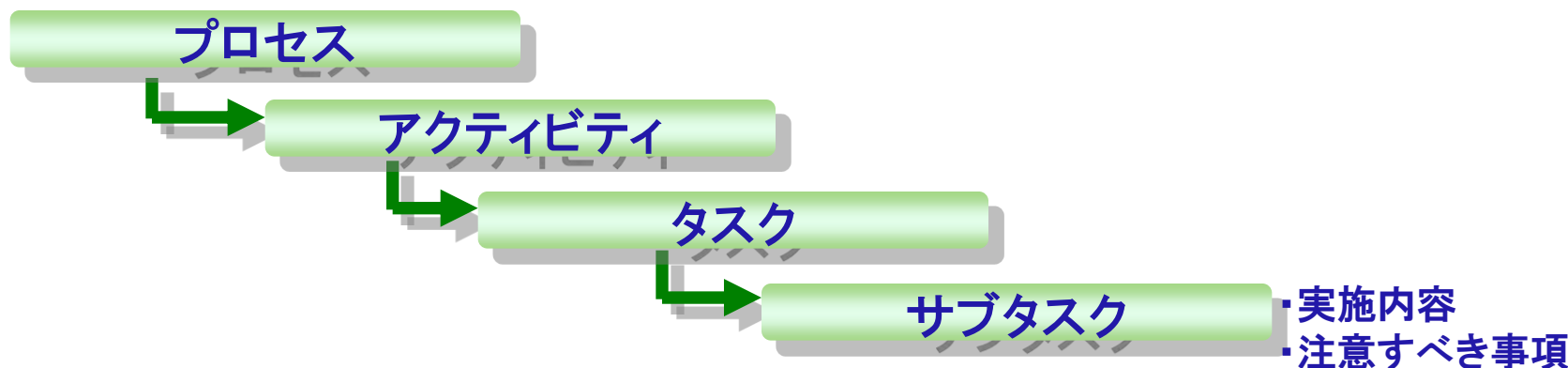
プロセス整備
工程設計
作業計画

プロセスの整理の考え方

○プロセス設計と工程設計を分離

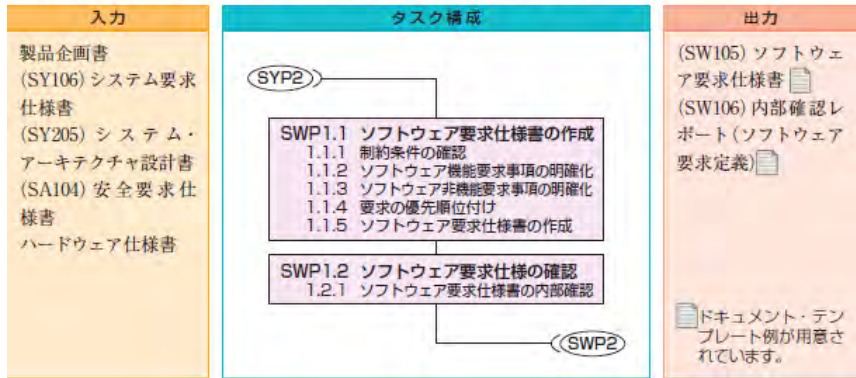
- ・実際の開発における時間軸と切り離し
  - （Ex、設計の繰り返し、テストファースト･･･）
- ・品質確保に必要な作業を整理
  - （上流工程の充実、レビュー/テストの重視）

○入出力物の明確化による開発手順を規定

○様々な作業をグルーピング化、階層的に整理

アクティビティの解説例

SWP1　ソフトウェア要求定義

当該製品を実現するためにソフトウェアとして実現が必要となる要求を明確にする。

入力	タスク構成	出力
製品企画書 (SY106) システム要求仕様書 (SY205) システム・アーキテクチャ設計書 (SA104) 安全要求仕様書 ハードウェア仕様書	(SYP2) SWP1.1 ソフトウェア要求仕様書の作成 1.1.1 制約条件の確認 1.1.2 ソフトウェア機能要求事項の明確化 1.1.3 ソフトウェア非機能要求事項の明確化 1.1.4 要求の優先順位付け 1.1.5 ソフトウェア要求仕様書の作成 SWP1.2 ソフトウェア要求仕様の確認 1.2.1 ソフトウェア要求仕様書の内部確認 (SWP2)	(SW105) ソフトウェア要求仕様書 (SW106) 内部確認レポート（ソフトウェア要求定義） ドキュメント・テンプレート例が用意されています。

■ 解説

このアクティビティは

(1.1.1)　システム要求仕様とハードウェア仕様をもとに、ソフトウェア要求仕様を検討する際の制約条件を確認した上で、

(1.1.2)　ソフトウェアに必要となる機能面の要求を明確にし、

(1.1.3)　また、ソフトウェアに必要な非機能面の要求も合わせて明確にする。

(1.1.4)　実際のソフトウェア開発では開発期間やリソース面の制約、あるいは、(1.1.1)で整理したシステムとしての制約などがあるため、これらを考慮したうえで、個々の機能、非機能要求に優先順位をつけていく。

(1.1.5)　これらの検討結果を整理し、ソフトウェア要求仕様書を作成する。

(1.2.1)　作成したソフトウェア要求仕様書はあらかじめ確認ポイントを決めて確認を行い、確認した内容を整理して内部確認レポートを作成する。

■ 留意事項

- ソフトウェア要求定義の開始条件として以下の事項に留意する。
  - ・製品企画：製品戦略（エンドユーザーニーズなど）が明確になっている
  - ・製品仕様：取扱説明書レベルの内容は決まっている
  - ・スケジュール：全体スケジュールは決まっている（発売日、対外的なマイルストーンなど）
  - ・システム・アーキテクチャ：以下の事項が明確になっている
ハードとソフトの機能分担（ソフトウェア要求は明確化されている）、ハードウェア構成、外部インタフェース、要求性能の実現方法、保守機能、セキュリティ
  - ・前提条件：使用するソフトウェア（OS、ライブラリなど）、既存製品の利用
- ソフトウェア要求定義にあたっては、システム要求定義で検討した以下の事項に留意する。
  - ・組込みシステムの場合、システムが動作する環境など外部環境の分析とそれに伴う異常処理対応の機能分析を考慮する
  - ・機能的な側面のみでなく性能や保守性など非機能的な側面を考慮する
  - ・対象システムの外部で連携動作するシステムにも留意する
  - ・製品として見た場合のロングレンジの製品機能のあり方も考慮する

■ [参考] 手法およびツール

- OOA（Object Oriented Analysis：オブジェクト指向分析）
- 構造化分析
- DFD（Data Flow Diagram：構造化分析で使われる分析技法）
- シナリオ分析
- プロトタイピング
- 品質機能展開

タスク/サブタスクの解説例

SWP1.1　ソフトウェア要求仕様書の作成

システム要求仕様書をもとにソフトウェアとして実現が求められる事項を明確にし、ソフトウェア要求仕様書としてまとめる。

1.1.1　制約条件の確認

入力	概要	出力
製品企画書 (SY106) システム要求仕様書 (SY205) システム・アーキテクチャ設計書 (SA104) 安全要求仕様書 各ハードウェア仕様書	ソフトウェアに関する要求事項を検討するにあたっての考慮すべき制約条件を明確にし、制約条件リストとしてまとめる。	(SW101) 制約条件リスト

参照情報
- ISO/IEC 9126-1*(ソフトウェア品質モデル)
- ISO 9241*(ユーザビリティ)

実施内容

以下に示す①〜⑥の事項を確認し、制約条件リストとして整理する。

①製品企画、製品開発戦略を確認する。ソフトウェア要求を定義するにあたって、考慮すべき製品目標を確認する。
- 特徴的な新機能の有無
- プロダクトライン開発の適用　など

②製品特性を確認する。
- 信頼性要求、安全性要求 安全
- 耐用年数、製品寿命
- 想定される利用状況、利用環境
- 準拠しなければならない規格・規約の有無など

③製品のステークホルダ***(利害関係者)を確認する。
- サービス部門、営業、企画、ハード開発部門、製造部門など、製品に関わるステークホルダを確認する。
- 製品のエンドユーザを明確にし、ユーザグループ別の特徴を確認する。
- ステークホルダ別に対応しなければならない制約条件をリストアップする。

④製品構成を確認する。
- ハードウェア構成とその制約。
- 利用するOS、ミドルウェアなどを明確にし、それぞれの制約をリストアップする。
- 製品が連携動作する周辺のソフトウェア、システムやハードウェアなどとのインタフェースを明確にする(センサー、アクチェータなど)。

⑤再利用ソフトウェアを確認する。
- 既存ソフトウェアを再利用するか否かを検討する。
- 再利用する場合は、再利用ソフトウェアの仕様や特徴、ならびに再利用の方針を確認する。

⑥ソフトウェアの開発環境、テスト環境、導入環境を確認する。
- 開発に利用するツール
- テスト環境、テスト用ツール、テスト方法、テストデータの利用可能性
- インストール時の制約なども明らかにしておく

注意すべき事項

●製品企画、製品開発戦略
- エンドユーザニーズ、製品バリエーションや中長期レンジのマーケット戦略などを確認する。
- 全体スケジュール(発売日、対外的なマイルストーンなど)を確認し、ソフトウェア開発に充当できる期間を明確にしておく。
- 競合製品に対して優位性を確保するための機能は何かについても確認する。
- ハードウェアプラットフォームやミドルウェア、周辺デバイスの技術進化とその時期などの製品戦略を確認する。
- 製品仕様(取扱説明書レベルの内容は決まっているか)を確認する。
- 達成すべき性能、機能について、TBD (To Be Determined：未定)事項などを明確にしておく。

●製品特性
- 信頼性：どの程度の信頼性が求められるかMTBF (Mean Time Between Failure：平均故障間隔)やMTTR (Mean Time To Repair：平均復旧時間)などの具体的な指標値を検討しておく。
- 動作環境：製品が利用されるコンテキストを明確にしておく(例：温度、ノイズ、静電気)。組込みシステムでは正常系のみでなく異常系の動作が発生しうるコンテキストについての調査も必須となる。
- 保守環境：保守のタイミングや方法についても検討しておく。
- 準拠規約：PL法、環境基準
- 安全性：システムに求められる安全に関する要求(安全度のレベルとそれを実現する機能など) 安全

●ステークホルダ
- ユーザについては、製品の一次ユーザのみだけでなく、その先の二次ユーザなども確認する。

●製品構成
- 使用するMPU/MCUの種類
- 利用可能なメモリ容量
- 入出力機器
- OS、ライブラリ
- ハードウェアとソフトウェアの機能分担

●再利用ソフトウェア
- ソフトウェアの再利用については、再利用の粒度に留意する(アーキテクチャの再利用なども視野に入れて考える)。

安全：セーフティに関連する作業

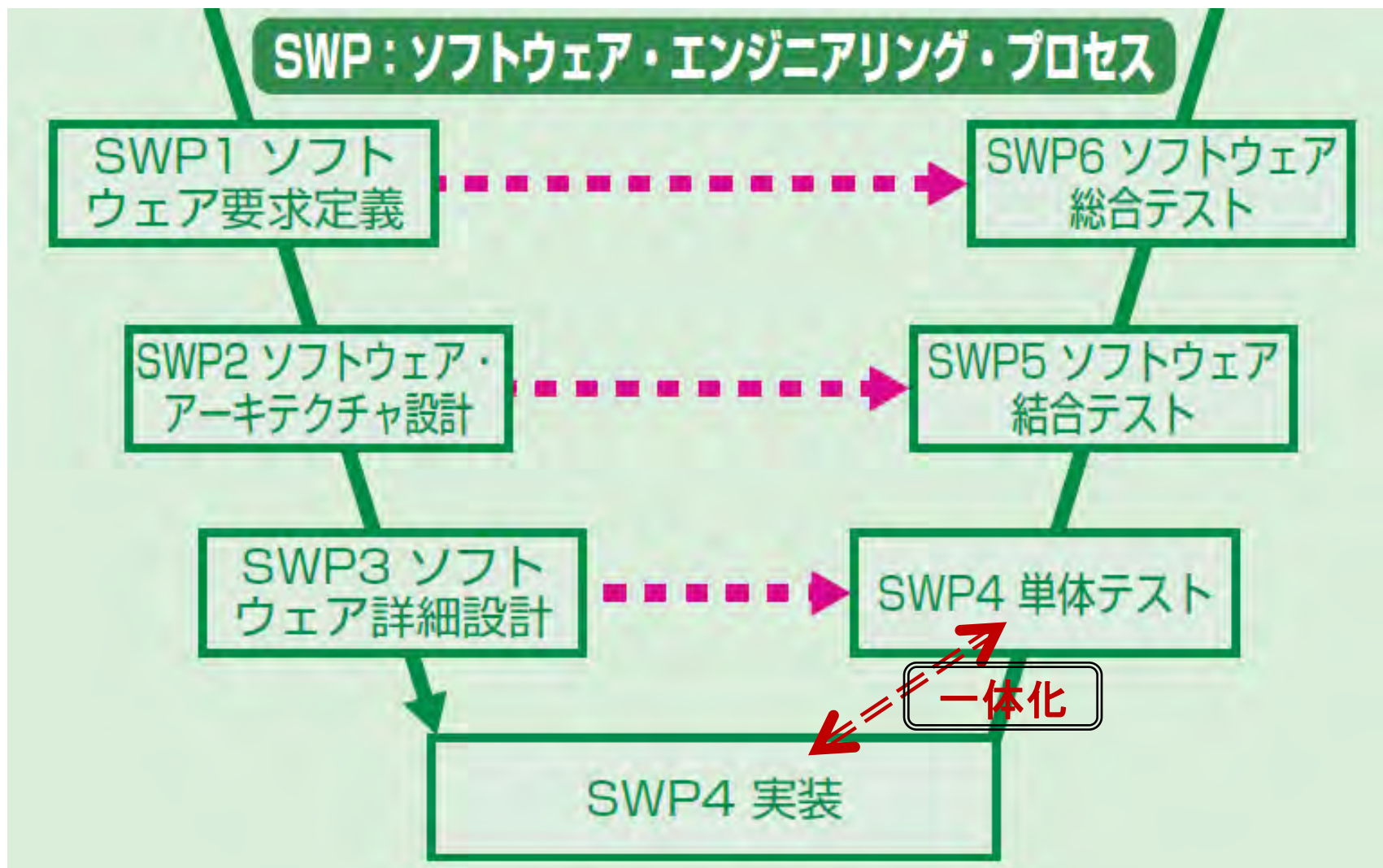
● 用語解説

* ISO/IEC 9126-1 (ソフトウェア品質モデル)、ISO 9241 (ユーザビリティ)、ISO 13407 (人間中心設計プロセス)：ソフトウェアの取り扱いの容易さや操作の分かりやすさなどの使い勝手 (ユーザビリティ) に関連する規格。

** IEC 61508 (安全性)：計算機などを内包し高い安全性を求められるシステム (プラント制御や自動車など) について、その開発過程で安全性実現の視点から実施すべき作業を整理した規格。

*** ステークホルダ：企業内からエンドユーザまで、製品に利害関係のある人のこと。

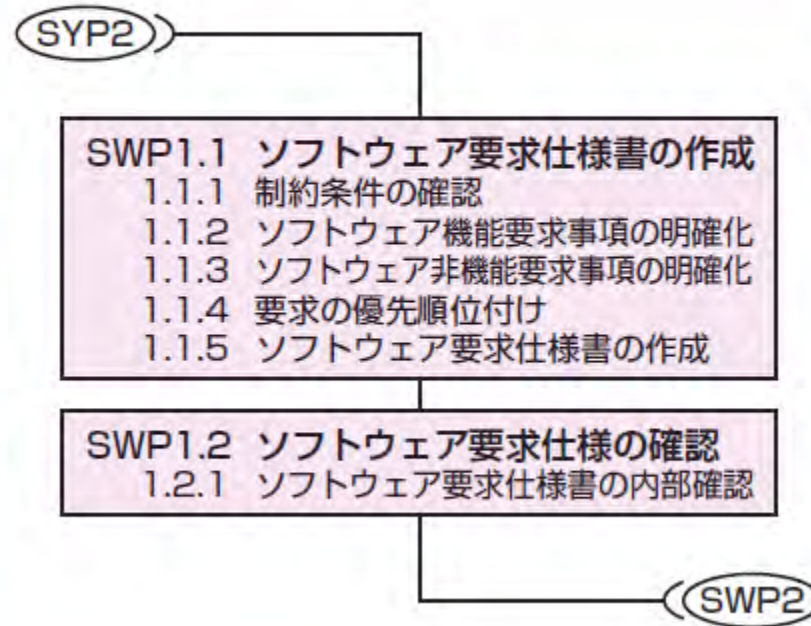
安全：セーフティに関連する作業

ソフトウェア･エンジニアリング･プロセス構成

ソフトウェア要求定義

（実現する要求事項の明確化）

入力	タスク構成	出力
製品企画書 (SY106) システム要求仕様書 (SY205) システム・アーキテクチャ設計書 (SA104) 安全要求仕様書 ハードウェア仕様書		

ソフトウェアアーキテクチャ設計

（アーキテクチャを決定し、アーキテクチャ設計書を作成）

入力	タスク構成	出力
(SW105) ソフトウェア要求仕様書 (SY205) システム・アーキテクチャ設計書 ハードウェア仕様書	SWP1 SWP2.1 ソフトウェア・アーキテクチャ設計書の作成 2.1.1 設計条件の確認 2.1.2 ソフトウェア構成の設計 2.1.3 ソフトウェア全体の振る舞いの設計 2.1.4 インタフェースの設計 2.1.5 性能/メモリ使用量の見積もり 2.1.6 ソフトウェア・アーキテクチャ設計書の作成 SWP2.2 ソフトウェア・アーキテクチャ設計の確認 2.2.1 ソフトウェア・アーキテクチャ設計書の内部確認 SWP2.3 ソフトウェア・アーキテクチャ設計の共同レビュー 2.3.1 ソフトウェア・アーキテクチャ設計書の共同レビュー SWP3	(SW205) ソフトウェア・アーキテクチャ設計書 (SW206) 内部確認レポート（ソフトウェア・アーキテクチャ設計） ドキュメント・テンプレート例が用意されています。

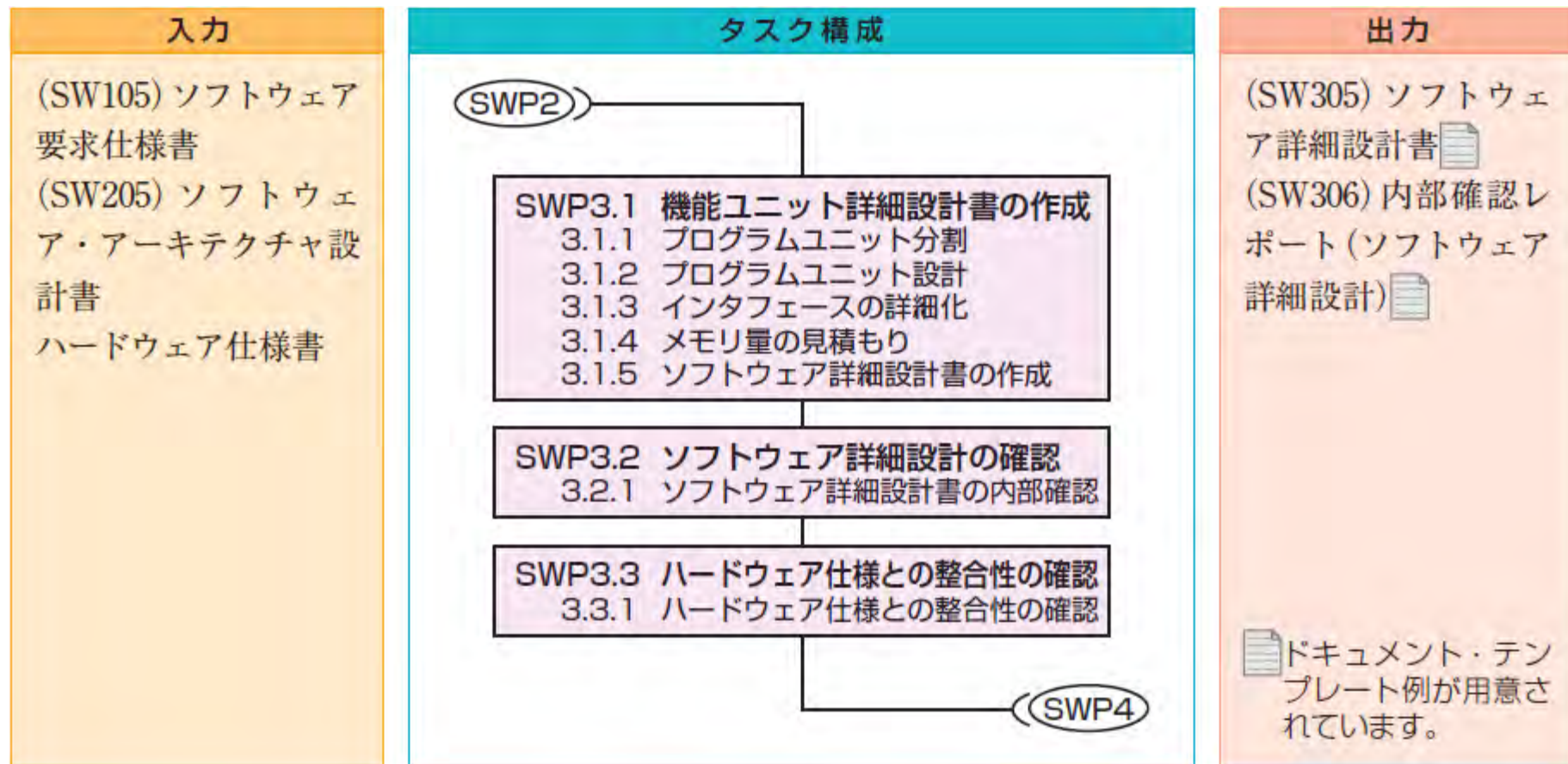
ソフトウェア詳細設計

（機能ユニットのプログラムユニット分割及び プログラムユニット詳細の設計と詳細設計書の作成）

入力

(SW105) ソフトウェア要求仕様書
(SW205) ソフトウェア・アーキテクチャ設計書
ハードウェア仕様書

タスク構成

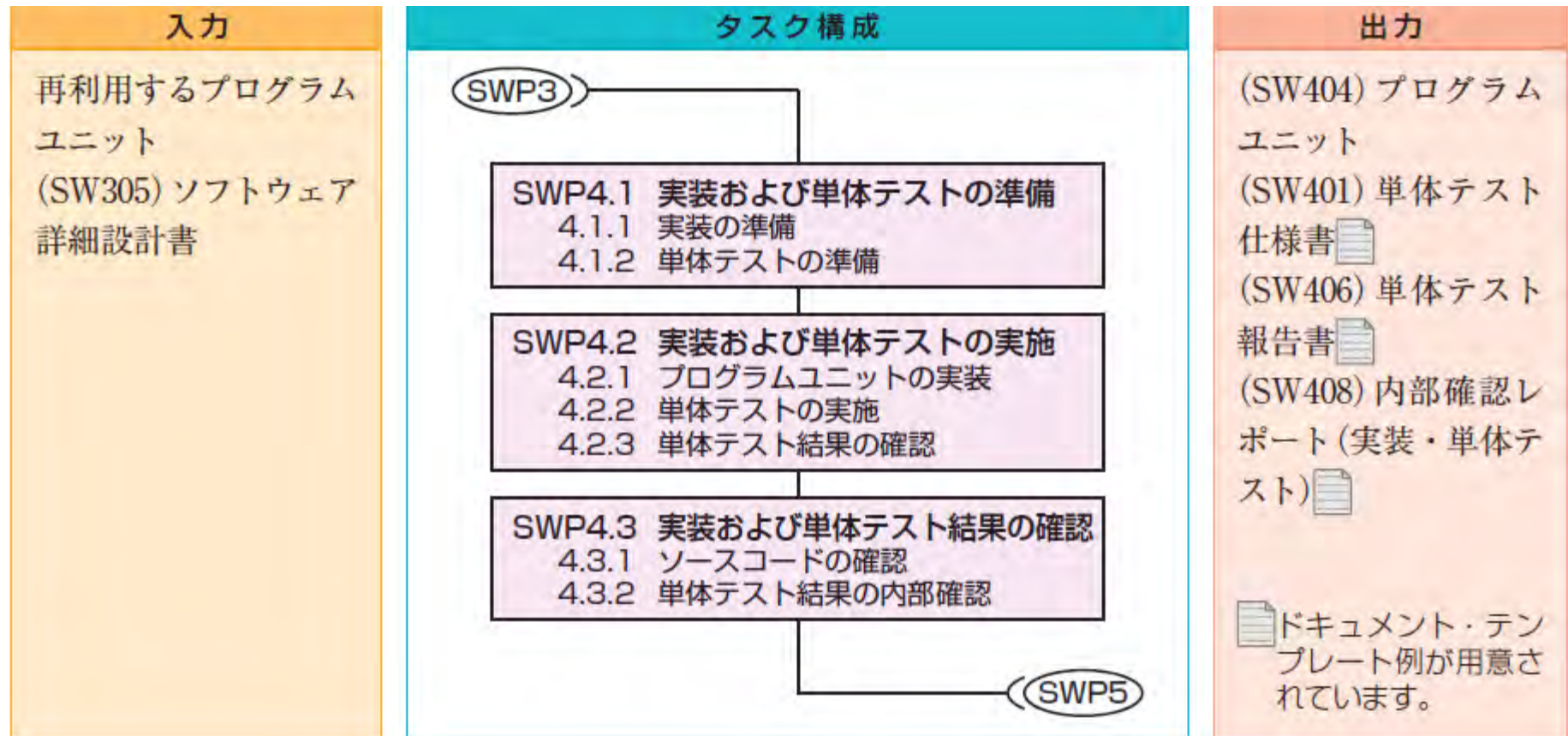
出力

実装および単体テスト

（プログラムユニットの実装と プログラムユニットの動作の確認）

入力

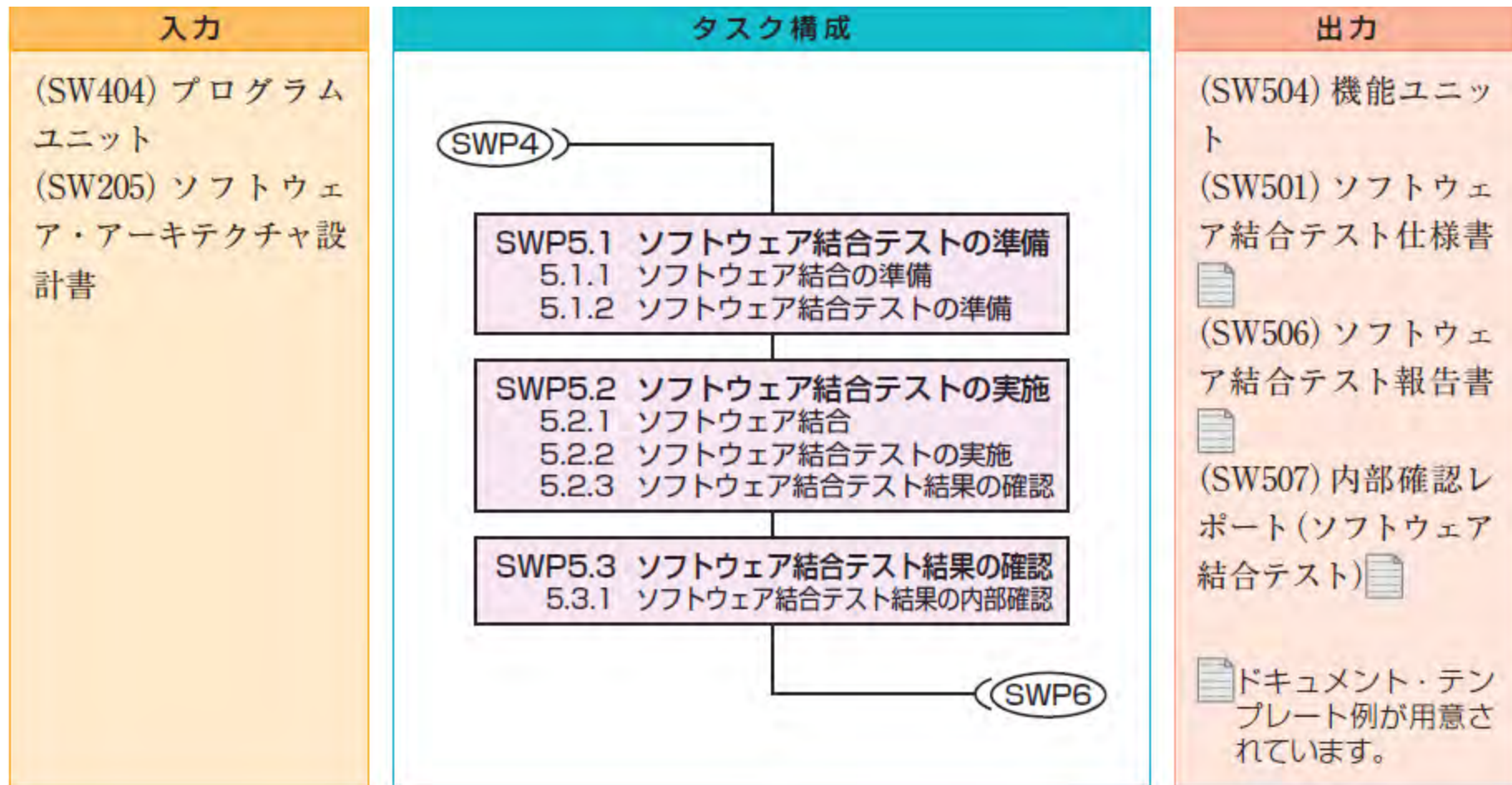
再利用するプログラムユニット
(SW305) ソフトウェア詳細設計書

タスク構成

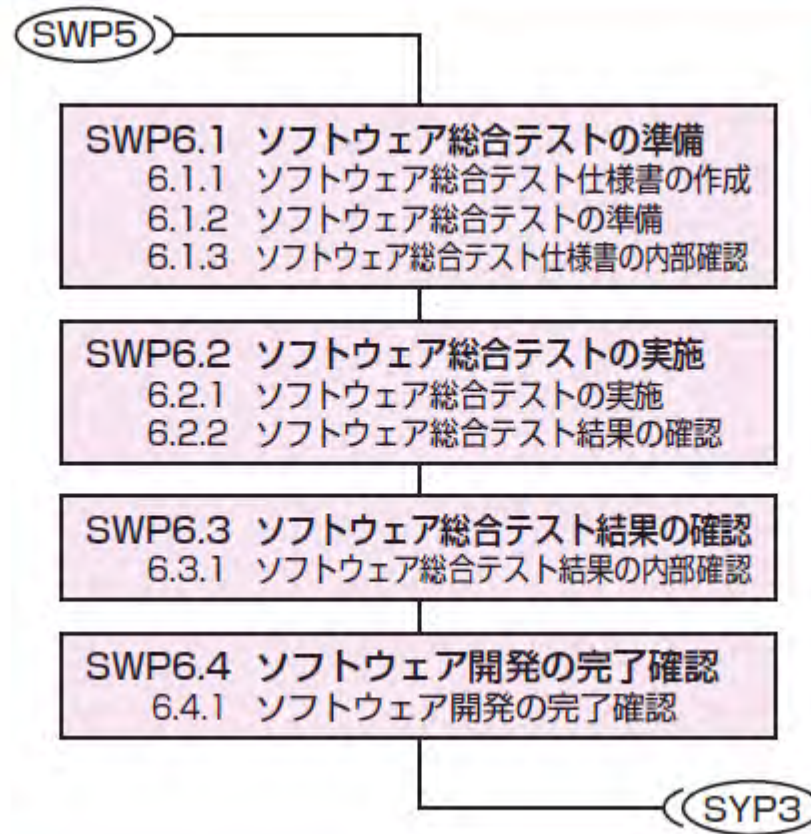
出力

ソフトウェア結合テスト

（プログラムユニットの組み立てと組立てた際の機能の動作を確認する）

入力

(SW404) プログラムユニット
(SW205) ソフトウェア・アーキテクチャ設計書

ソフトウェア総合テスト

（全機能ユニットを結合した状態で、ソフトウェアとしての総合的な動作の確認）

(SW504) 機能ユニット
(SW105) ソフトウェア要求仕様書

(SW601) ソフトウェア総合テスト仕様書

(SW606) ソフトウェア総合テスト報告書

(SU103) プロジェクト完了報告書(ソフトウェア開発)

ドキュメント・テンプレート例が用意されています。

ドキュメント・テンプレート

ドキュメント・テンプレート例

- （SY106）システム要求仕様書
- （SY205）システム・アーキテクチャ設計書
- （SA104）安全要求仕様書
- （SW105）ソフトウェア要求仕様書
- （SW205）ソフトウェア・アーキテクチャ設計書
- （SW305）ソフトウェア詳細設計書
- テスト仕様書／テスト報告書
  - （SW401）単体テスト仕様書
  - （SW501）ソフトウェア結合テスト仕様書
  - （SW601）ソフトウェア総合テスト仕様書
  - （SY301）システム結合テスト仕様書
  - （SY401）システムテスト仕様書
  - （SA201）安全性テスト仕様書
  - （SW406）単体テスト報告書
  - （SW506）ソフトウェア結合テスト報告書
  - （SW606）ソフトウェア総合テスト報告書
  - （SY306）システム結合テスト報告書
  - （SY406）システムテスト報告書
  - （SA205）安全性テスト報告書
- 内部確認レポート
  - （SY107）内部確認レポート（システム要求仕様）
  - （SY206）内部確認レポート（システム・アーキテクチャ設計）
  - （SA105）内部確認レポート（安全要求仕様）
  - （SW106）内部確認レポート（ソフトウェア要求定義）
  - （SW206）内部確認レポート（ソフトウェア・アーキテクチャ設計）
  - （SW306）内部確認レポート（ソフトウェア詳細設計）
  - （SW408）内部確認レポート（実装・単体テスト）
  - （SW507）内部確認レポート（ソフトウェア結合テスト）
  - （SW607）内部確認レポート（ソフトウェア総合テスト）
  - （SY307）内部確認レポート（システム結合テスト）
  - （SY407）内部確認レポート（システムテスト）
  - （SA206）内部確認レポート（安全性テスト）
- （SU801）共同レビュー記録
  - （SU801）共同レビュー記録（システム・アーキテクチャ設計）
  - （SU801）共同レビュー記録（ソフトウェア・アーキテクチャ設計）
  - （SU801）共同レビュー記録（ソフトウェア総合テスト）
  - （SU801）共同レビュー記録（システムテスト）
- （SU601）不具合管理票
- （SU602）不具合管理台帳
- プロジェクト完了報告書
  - （SU103）プロジェクト完了報告書（ソフトウェア開発）
  - （SU104）プロジェクト完了報告書（システム開発）

＊典型的なドキュメントについてテンプレートを提供

ドキュメント・テンプレート例

■（SW205）ソフトウェア・アーキテクチャ設計書の例

表　紙

文書名 — 文書名を記載する。

文書番号 — 文書の識別情報を記載する。

承認	作成
承認者名	作成者名
承認日	作成日

作成・承認の責任・確認を記載する欄を設ける。

発行日
発行部署 — 発行組織名、発行日を記載する。

改訂履歴

文書名

改訂履歴

項番	日付	バージョン	改訂内容	備考
1				
2				
3				
4				

改訂情報を記載できる欄を設ける。

目　次

文書名

目　次

文書番号　頁番号　発行日・発行部署

本文①

文書名

1. 概要

<記載例>
・本書の目的
・本書の位置づけ
・記載範囲、記載内容など
・参照しているドキュメントなど
・定義（用語、略語など）

ドキュメントの目的、位置づけ、記載内容などの本書の概要および参照しているドキュメント名等を記載する。

2. システム構成

<記載例>
・システム全体構成
－システム構成要素の名称/基本機能
－ソフトウェアに関係する外部要素の基本仕様

例：MPU名称、動作周波数、ハード割込みレベルとI/Oの対応、メモリー種類とサイズ、通信プロトコル、入出力データ形式、外部信号、外部機器制御方式など

ハードウェアを含めたシステム全体の構成とソフトウェアの位置づけ、およびソフトウェアを取り巻く関係/条件を記載する。
システム・アーキテクチャ設計書、ハードウェア仕様書等の関連資料より要点、条件などを整理して記載する。

3. ソフトウェア構成

<記載例>
・ソフトウェア全体構成
－ソフトウェア機能ユニットの名称/基本機能/動作仕様
（動作仕様とは、タスク/非タスク、タスクプライオリティ、システム定義/ユーザ定義、周期/非周期、共有資源、使用資源など）
－ソフトウェア機能ユニット間のインタフェース

例：各機能ユニット間の関係、データを静的に記述したもので、必要に応じて、OS、メモリ、外部記憶媒体、ハードウェア機構を明記する。

開発ソフトウェアの構成および基本的関係を記載する。

4. 制御方式

ソフトウェア要求仕様書（SW105）に記載された要求をシステム全体（ハードウェア、オペレータ、外部記憶媒体、メモリなど）を通してどのように実現するかを、ソフトウェア機能ユニットを中心に記載する。

4.1 メモリー構成/レイアウト

<記載例>
・メモリアドレス空間を基に、ROM/RAMの種類およびSRAM/キャッシュ割り当て領域、ソフトウェア領域、ハードウェアレジスタ領域、サイズなど
・ソフトウェア領域の構成（コード域、データ域、OS使用域）
・ソフトウェア共通領域の構成および、テーブル/バッファのリンク方式など

メモリ構成、種類、サイズ等、システム・アーキテクチャ設計書または、ハードウェア仕様書等から明確にするとともにソフトウェア領域をどのように使用するかを記載する。
（ただし、高機能32ビットMPU、VM機能付きOS、SoC等の場合は、固定化あるいはバーチャル化されており、ソフトウェアでの必要な事項に関して明確化すれば良い。）

4.2 ソフトウェア制御方式

<記載例>
・リカバリ方式：エラー発生から再立ち上げまでの制御の流れ
・性能実現方式：要求される性能を実現する並行処理制御
・制御方式：複数の外部要因に対処するための優先方式
・出荷後のソフトウェアアップデート方式：フィールド出荷後のソフトウェア更新方式
・障害調査データ収集方式：事件トレースなどの収集から取得/解析までの流れなど

ソフトウェア機能ユニット間の処理シーケンス図等を利用し、信号、コマンド、データ等の検知、制御、変換、処理、通知、格納等を、ケースごとに記載する。また、要求としては明に記載されないが、開発者の共有事項として重要な事項（たとえば、OS動作、内部状態遷移）についても記載するのが望ましい。

4.3 性能見積

<記載例>
・必要性能/制限：x x [ms]
・性能実現方式：対象条件とその処理シーケンス
・外部接続処理見積もり：外部接続要素の見積もりおよび根拠

定量的に性能を見積もる。

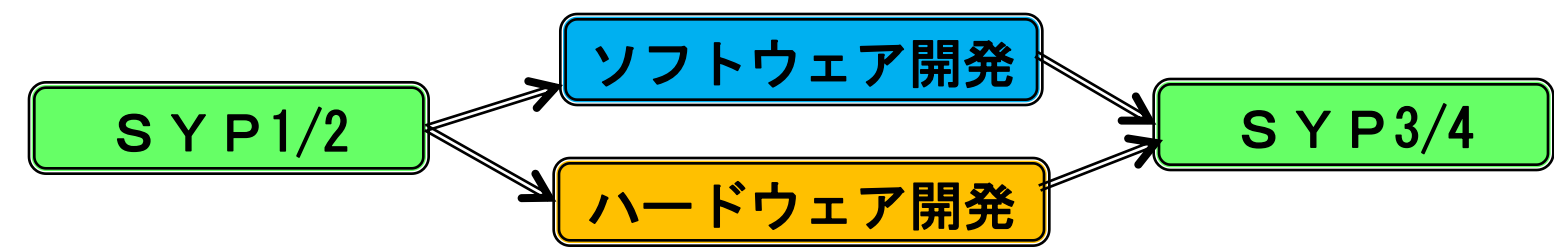
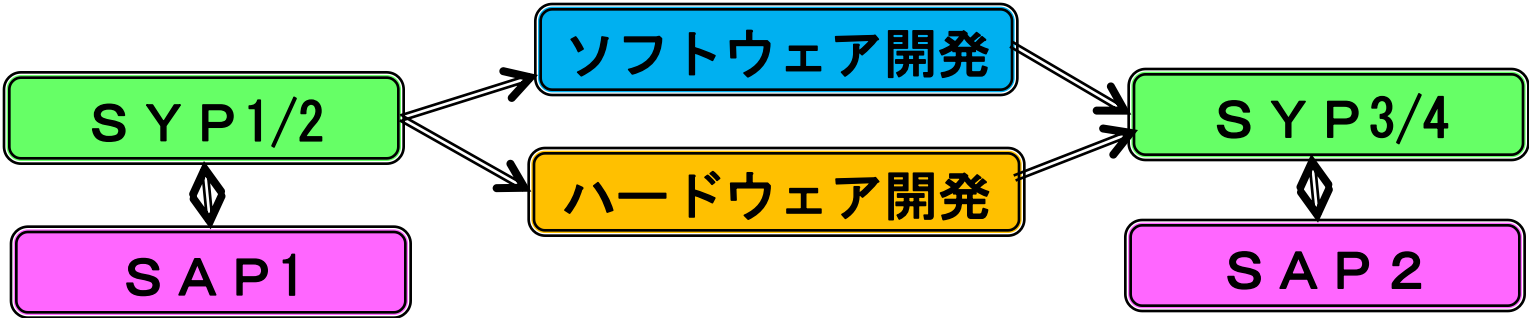
文書番号　頁番号　発行日・発行部署

システム・エンジニアリング・プロセス

SEC
Software Engineering
for Mo･No･Zu･Ku･Ri

ID	アクティビティ	アクティビティの概要	構成するタスク
SYP1	システム要求定義	当該製品を実現するためにシステムとして実現が必要となる要求を明確にする。	SYP1.1　システム要求仕様書の作成 SYP1.2　システム要求仕様の確認
SYP2	システム・アーキテクチャ設計	開発する組込みシステムをどのように実現するか、ハード/ソフトの役割分担も含めて明確にする。	SYP2.1　システム・アーキテクチャ設計書の作成 SYP2.2　システム・アーキテクチャ設計の確認 SYP2.3　システム・アーキテクチャ設計の共同レビュー
SYP3	システム結合テスト	システムを構成するハードとソフトを組み合わせた際の機能ブロックが動作することを確認する。	SYP3.1　システム結合テストの準備 SYP3.2　システム結合テストの実施 SYP3.3　システム結合テスト結果の確認
SYP4	システムテスト	システムの要求事項が実現できていることを確認する。	SYP4.1　システムテストの準備 SYP4.2　システムテストの実施 SYP4.3　システムテスト結果の確認 SYP4.4　システム開発の完了確認

セーフティ・エンジニアリング・プロセス

SEC Software Engineering for Mo·No·Zu·Ku·Ri

ID	アクティビティ	アクティビティの概要	構成するタスク
SAP1	安全性要求定義	当該製品に関して、安全性に関する要求事項を洗い出し、安全要求仕様書に整理する。	SAP1.1 安全要求仕様書の作成 SAP1.2 安全要求仕様の確認
SAP2	安全性テスト	開発した製品に関して、安全性の視点からのテストを実施する。	SAP2.1 安全性テストの準備 SAP2.2 安全性テストの実施 SAP2.3 安全性テスト結果の確認

サポート・プロセス

ID	アクティビティ	アクティビティの概要	構成するタスク
SUP1	プロジェクトマネジメント	組込みソフトウェアを開発するための開発プロジェクトを定義し、そのプロジェクトの活動を円滑に進めるための作業を規定する。	SUP1.1 プロジェクト計画書作成 SUP1.2 プロジェクト実施状況の把握 SUP1.3 プロジェクトの制御 SUP1.4 プロジェクト完了報告書の作成
SUP2	品質保証	開発している組込みソフトウェアの品質が要求や市場ニーズに合致するように、開発過程での品質の作りこみを実現するための作業を規定する。	SUP2.1 品質目標の設定 SUP2.2 品質保証方式の確定 SUP2.3 品質の可視化に基づく品質制御
SUP3	リスクマネジメント	組込みソフトウェアの開発過程で発生	SUP3.1 リスクの洗い出しと把握
SUP4	文書化と文書管理		
SUP5	構成管理		

ID	アクティビティ	アクティビティの概要	構成するタスク
SUP6	問題解決管理	開発の過程で生ずるさまざまな問題点や課題を把握し、それらへの対策や解決状況を管理する。	SUP6.1 問題の記録と原因分析 SUP6.2 影響分析と対策立案 SUP6.3 対策の実行 SUP6.4 対策結果の確認
SUP7	変更管理	開発着手後に発生する要求や設計の変更とそれらへの対応を管理する。	SUP7.1 変更要求情報の記録 SUP7.2 変更による影響の分析 SUP7.3 変更計画の立案と実施 SUP7.4 変更結果の確認
SUP8	共同レビュー	開発作業の節目ごとに当該プロセスでの作業結果が適切であったかどうかを、関係者間で技術面、管理面の両面から確認する。	SUP8.1 レビューの準備 SUP8.2 レビューの実施 SUP8.3 レビュー結果の確認と
SUP9	開発委託管理	一部のプロセスを外部委託する際に必要となる作業を規定する。	SUP9.1 発注の準備と契約 SUP9.2 開発委託作業のモニタ
SUP10	開発環境整備	設計から、実行モジュール作成、テストに到るまでの開発に必要となる環境(実装環境、テスト環境など)を整備し管理する。	SUP10.1 開発環境整備計画の立 SUP10.2 開発環境の構築 SUP10.3 開発環境の維持

標準プロセスとしての活用

問題把握と対策

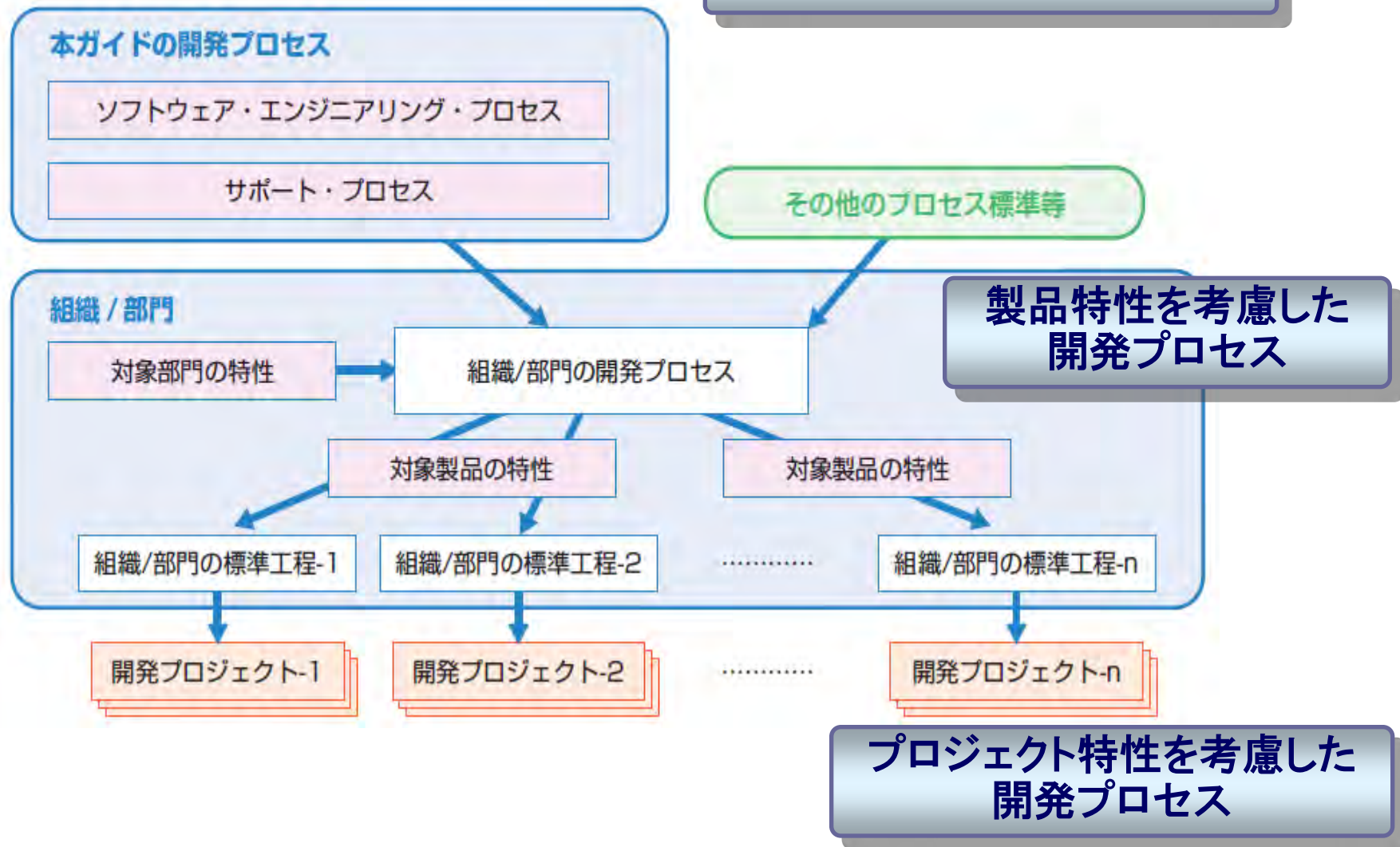
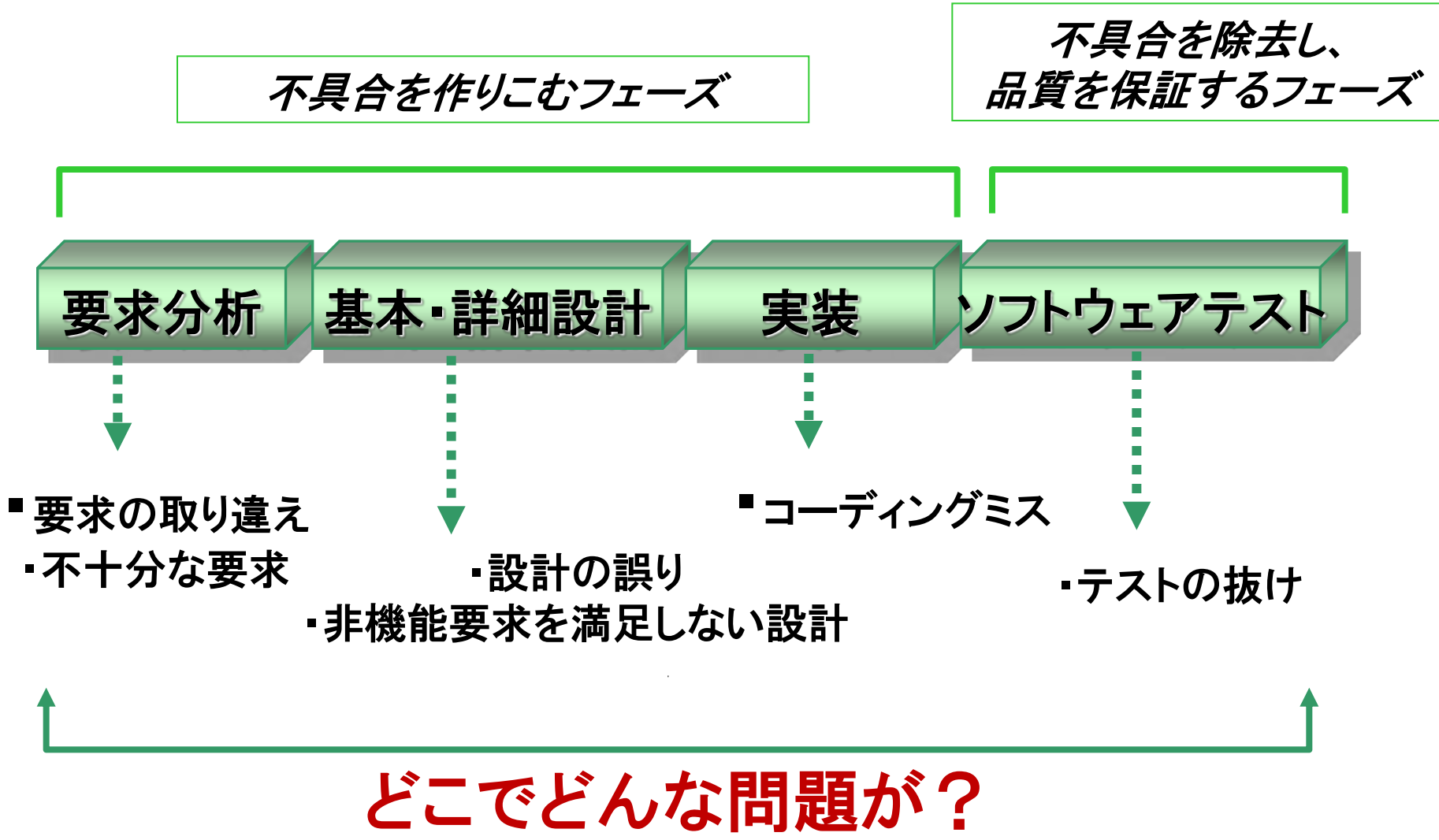
不具合を除去し、品質を保証するフェーズ

要求分析 基本･詳細設計 実装 ソフトウェアテスト

- 要求の取り違え
- ･不十分な要求
- ･設計の誤り
- ･非機能要求を満足しない設計
- コーディングミス
- ･テストの抜け

どこでどんな問題が？

テーラリングとプロセス(作業)の再起動

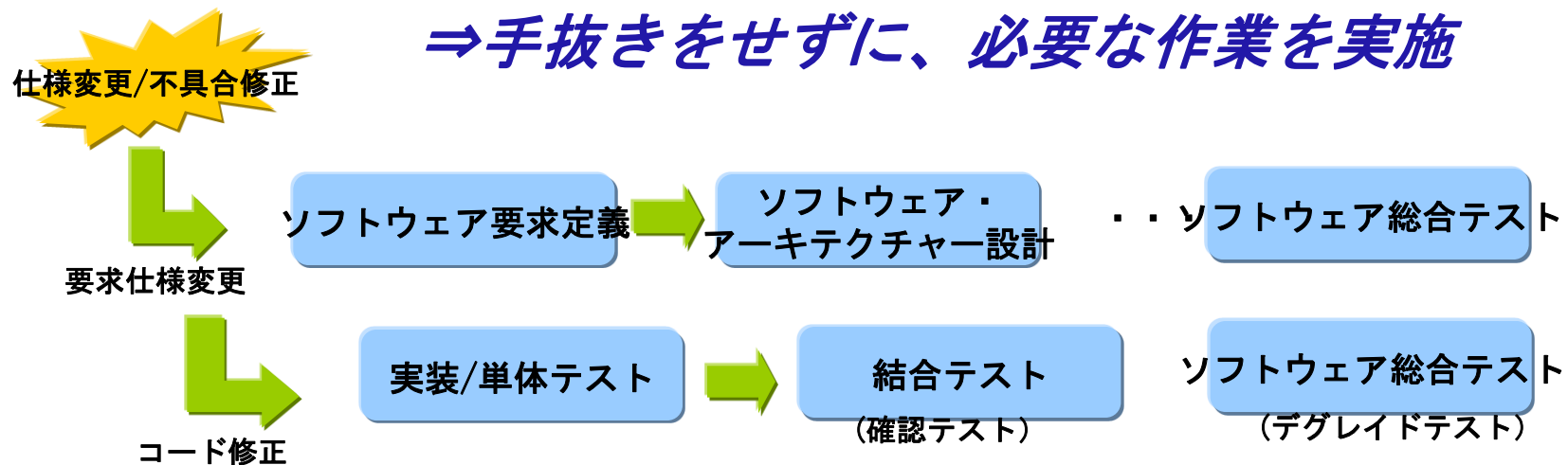
SEC
Software Engineering for Mo·No·Zu·Ku·Ri

■ 企業/組織に合ったプロセス

●関係部署、作業内容/名称、
ﾄﾞｷｭﾒﾝﾄ内容/名称など多種多様

⇒品質確保/効率化の要点を抽出してテーラリング

■ 修正/変更後のプロセス(作業)再起動

●仕様変更/不具合修正は必ずある。

⇒手抜きをせずに、必要な作業を実施

さまざまな開発形態（１）

■ 事例１　ハード環境構築が困難なテスト環境

- ●テスト機の機能制限、品質不良、提供遅れ、台数不足･･･

■ 事例2　小規模で、短期間の開発

- ●標準の開発手順を踏めなく、十分なﾚﾋﾞｭｰ/ﾄﾞｷｭﾒﾝﾄ作成が困難

■ 事例3　事前の作業が必要な開発

- ●試作ハードの評価、ソフト処理性能確認が実開発の前に必要

■ 事例4　高効率を目指した開発

- ●高い流用/再利用率、自動コード生成率

■ 事例5　新規要素の高い開発

- ●スパイラル開発、未決定仕様が多く仕様変更が予想される開発

■ 事例6　外部委託開発が多い開発

- ●新規の委託先、海外委託、特定工程作業の委託

■ 事例7　特定の作業のみの開発

- ●ｺｰﾃﾞｨﾝｸﾞ、ﾃｽﾄ作業のみの実施

プロダクト vs. プロセス

プロダクト品質　：製品としてのソフトウェアが有する品質

(バグが無いこと使いやすいこと)

ソフトウェアそのもの
- 内部構造
- 動き/振舞い

プロダクト
開発途中で作られる**中間成果物**
開発の結果得られる**最終成果物**

開発の過程
- PJの全貌
- 開発の状況

プロセス
開発で実施される作業

プロセス品質　：ソフトウェア開発作業の品質

(必要な手順を踏んで開発されていること)

開発プロセス整備で得られる効果

作業漏れの回避、非効率作業の改善

品質作りこみ仕組みの確立

作業分担や作業間連携の明確化

ﾌﾟﾛｾｽ情報共有によるｽﾑｰｽﾞな開発委託

ご清聴ありがとうございました